Vybrid Reference Manual

F-Series

Document Number: VYBRIDRM
Rev. 5, 07/2013

freescale™

Contents

Section number **Title** **Page**

Chapter 1
About This Document

Chapter 2
Overview

Section number	Title	Page

Chapter 3
Memory Map

Chapter 4
Signal Multiplexing

Chapter 5
Input/Output Multiplexer Controller (IOMUXC)

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Chapter 6
Port control and interrupts (PORT)

Section number	Title	Page

Chapter 7
General-Purpose Input/Output (GPIO)

Chapter 8
Chip Configuration

Section number	Title	Page

Chapter 9

Section number	Title	Page

Section number	Title	Page

Chapter 10
Clock Controller Module (CCM)

Chapter 11
Analog components control digital interface (ANADIG)

Chapter 12
Slow Clock Source Controller Module (SCSC)

Chapter 13
Clock Monitor Unit (CMU)

Section number	Title	Page

Chapter 14
Power management

Chapter 15
Global Power Controller (GPC)

Section number	Title	Page

Chapter 16
Voltage Regulators

Chapter 17
Reset

Chapter 18
System Reset Controller (SRC)

Section number	Title	Page

Chapter 19
System Boot

Section number	Title	Page

Chapter 20 Debug Architecture

Section number	Title	Page

Chapter 21
Direct Memory Access Controller (eDMA)

Section number	Title	Page

Chapter 22
Direct Memory Access Multiplexer (DMAMUX)

Chapter 23
Peripheral Bridge (AIPS-Lite)

Chapter 24
IPS_Semaphores

Section number	Title	Page

Chapter 25
External Watchdog Monitor (EWM)

Chapter 26
Watchdog Timer (WDOG)

Section number	Title	Page

Chapter 27
Watchdog Configuration

Chapter 28
Wakeup Unit (WKPU)

Section number	Title	Page

Chapter 29
Local Memory Controller

Section number | Title | Page

Chapter 30
Quad Serial Peripheral Interface (QuadSPI)

Section number	Title	Page

Chapter 32
External Bus Interface (FlexBus)

Chapter 33
Cyclic Redundancy Check (CRC)

Chapter 34
LPDDR2/DDR3 SDRAM Memory Controller (DDRMC)

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Chapter 35
On-Chip One Time Programmable (OCOTP) Controller

Section number	Title	Page

Chapter 36
12-bit Digital-to-Analog Converter (DAC)

Chapter 37
ADC-Digital-12b-1MSPS-SAR

Section number	Title	Page

Chapter 38
Programmable Delay Block (PDB)

Section number	Title	Page

Chapter 39
FlexTimer Module (FTM)

Chapter 40
Periodic Interrupt Timer (PIT)

Section number	Title	Page

Chapter 41
Low-Power Timer (LPTMR)

Section number	Title	Page

Chapter 42
10/100-Mbps Ethernet MAC (ENET)

Section number	Title	Page

Section number	Title	Page

Chapter 43
Ethernet Switch (ESW)

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Chapter 44
Universal Serial Bus (USB) Controller

Section number	Title	Page

Chapter 45
Universal Serial Bus 2.0 Integrated PHY (USBPHY)

Chapter 46
CAN (FlexCAN)

Section number	Title	Page

Chapter 47
Serial Peripheral Interface (SPI)

Chapter 48
Inter-Integrated Circuit (I2C)

Chapter 49
Universal Asynchronous Receiver/Transmitter (UART)

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Chapter 51
Integrated Interchip Sound (I2S) / Synchronous Audio Interface (SAI)

Section number	Title	Page

Chapter 52
Enhanced Serial Audio Interface (ESAI)

Section number	Title	Page

Section number	Title	Page

Chapter 53
Asynchronous Sample Rate Converter (ASRC)

Chapter 54
Sony/Philips Digital Interface (SPDIF)

Section number	Title	Page

Section number	Title	Page

Section number	Title	Page

Chapter 56
LCD Driver LCD64F6B

Section number	Title	Page

Chapter 57
Video-In VIU3

Section number	Title	Page

Section number	Title	Page

Chapter 58
Timing Controller TCON

Section number	Title	Page

Section number	Title	Page

Chapter 59
Run Length Encoding Decoder RLE_DEC

Chapter 60 Video subsytem

Section number	Title	Page

Section number	Title	Page

Chapter 1
About This Document

1.1 Overview

1.1.1 Purpose

This document describes the features, architecture, and programming model of the Freescale Vybrid microprocessor (MPU).

1.1.2 Audience

This document is primarily for system architects and software application developers who are using or considering using this device in a system.

1.1.3 Related Resources

Type	Description	Resource
Data Sheet	The Data Sheet includes electrical characteristics and signal connections.	VYBRIDFSERIESEC[1] -, [1]
Reference Manual	The Reference Manual contains a comprehensive description of the structure and function (operation) of the device.	VYBRIDFSERIESRM[1]
Chip Errata	The chip mask set Errata provides additional or corrective information for a particular device mask set.	Vybrid Chip Errata[1]

Table continues on the next page...

Type	Description	Resource
Application Notes	Application Notes are engineering support documents that assist the user in evaluating the operation of a device product line, package type, or general application topic.	• AN4651 [1] - TFT Panel Support in the Vybrid Microcontroller Family • AN4672 [1] - Using the Run-Length Decoding Features on Vybrid Devices • AN4647 [1] - Configuring and Using the 2D-ACE on Vybrid Microcontrollers • AN4635 [1] - Using the Vybrid TCON Module • AN4512 [1] - Quad Serial Peripheral Interface (QuadSPI) Module Updates

1. To find the associated resource, go to www.freescale.com and perform a search using this term.

1.2 Conventions

1.2.1 Numbering systems

The following suffixes identify different numbering systems:

This suffix	Identifies a
b	Binary number. For example, the binary equivalent of the number 5 is written 101b. In some cases, binary numbers are shown with the prefix *0b*.
d	Decimal number. Decimal numbers are followed by this suffix only when the possibility of confusion exists. In general, decimal numbers are shown without a suffix.
h	Hexadecimal number. For example, the hexadecimal equivalent of the number 60 is written 3Ch. In some cases, hexadecimal numbers are shown with the prefix *0x*.

1.2.2 Typographic notation

The following typographic notation is used throughout this document:

Example	Description
placeholder, x	Items in italics are placeholders for information that you provide. Italicized text is also used for the titles of publications and for emphasis. Plain lowercase letters are also used as placeholders for single letters and numbers.

Table continues on the next page...

Example	Description
`code`	Fixed-width type indicates text that must be typed exactly as shown. It is used for instruction mnemonics, directives, symbols, subcommands, parameters, and operators. Fixed-width type is also used for example code. Instruction mnemonics and directives in text and tables are shown in all caps; for example, BSR.
SR[SCM]	A mnemonic in brackets represents a named field in a register. This example refers to the Scaling Mode (SCM) field in the Status Register (SR).
REVNO[6:4], XAD[7:0]	Numbers in brackets and separated by a colon represent either: • A subset of a register's named field For example, REVNO[6:4] refers to bits 6–4 that are part of the COREREV field that occupies bits 6–0 of the REVNO register. • A continuous range of individual signals of a bus For example, XAD[7:0] refers to signals 7–0 of the XAD bus.

1.2.3 Special terms

The following terms have special meanings:

Term	Meaning
asserted	Refers to the state of a signal as follows: • An active-high signal is asserted when high (1). • An active-low signal is asserted when low (0).
deasserted	Refers to the state of a signal as follows: • An active-high signal is deasserted when low (0). • An active-low signal is deasserted when high (1). In some cases, deasserted signals are described as *negated*.
reserved	Refers to a memory space, register, or field that is either reserved for future use or for which, when written to, the module or chip behavior is unpredictable.
w1c	Write 1 to clear: Refers to a register bitfield that must be written as 1 to be "cleared."

Chapter 2
Overview

2.1 Vybrid Platform

This family of devices is Freescale's latest Dual Core offering with ARM® Cortex®-A5 and ARM® Cortex®-M4 based processors for high end industrial and general purpose applications.

NOTE

Throughout this manual, ARM® Cortex®-A5 is referred as Cortex-A5 or CA5. Also, ARM® Cortex®-M4 is referred as Cortex-M4 or CM4.

These devices are highly integrated reducing system cost for target applications. Main features include

- Cortex-A5 @500MHz (1.57 DMIPS/MHz) with TrustZone with 32 KB I-Cache/32 KB D-Cache
- Neon Media Processing Engine (MPE) co-processor and double precision Floating Point Unit (FPU)
- Cortex-M4 @167 MHz with 16 KB I-Cache/16 KB D-Cache
- 1.5 MB on-chip SRAM with ECC on 512 KB SRAM
- Support for LPDDR2/DDR3
- Dual TFT display up to SVGA and optional 40x4 and 38x6 Segmented LCD
- Dual 10/100 Ethernet with on-chip L2 Switch
- Dual USB OTG with on-chip HS PHY and on-chip HS/FS/LS PHY
- Advanced Security supporting Symmetric with on-chip Tamper detection
- Rich set of communication peripherals and general purpose features
- Advanced digital audio support with multiple audio interfaces and hardware asynchronous sample-rate converter co-processor.
- Multiple package options that include 176 LQFP, and 364 BGA

This family of devices is manufactured utilizing 40nm low-power process.

2.2 Feature Set

This family of devices supports the following features:

Table 2-1. Feature Set

Operating Characteristics	• Voltage range 3.0 V - 3.6 V • Temperature range (T_J) -40 to 105 °C • Flexible modes of operation
ARM Cortex-A5 Core	• Up to 500 MHz ARM Cortex-A5 • 32 KB/32 KB I-/D- L1 Cache • 1.57 DMIPS/MHz based on ARMv7 architecture • NEON MPE co-processor • Dual precision FPU • Optional 512K L2 Cache
ARM Cortex-M4 Core	• Up to 167 MHz ARM Cortex M4 • Build-in DSP capability • 16KB/16KB I-/D- L1 Cache • 1.25 DMIPS/MHz based on ARMv7 architecture
Clocks	• 6 Phase Locked Loops (PLLs) • 2 external crystal ocillators (XOSC) • 2 internal RC (IRC) oscillators
System, protection and power management features	• Flexible stop, wait, and run modes to provide lower power based on application needs. • Peripheral clock enable register can disable clocks to unused modules, thereby reducing currents • Low voltage warning and detect • Hardware CRC module to support fast cyclic redundancy checks (CRC) • 128-bit unique chip identifier • Hardware watchdog • External Watchdog Monitor (EWM) • Dual DMA controller with 32 channels (with DMAMUX)
Debug	• Standard JTAG • 16 bit Trace port
Timers	• Low Power Timer (LPTMR) • One Periodic Interrupt Timer (PIT) with 8 channels • IEEE® 1588 Timers (part of Ethernet Subsystem)
Communications	• UART(w/ LIN2.0, ISO7816, IrDA, LON and hardware flow control) • Serial peripheral interface (SPI) • I^2C with SMBUS support • Dual USB 2.0 HS OTG Controller (Supports LS/FS/HS) with integrated PHYs • 4/8 bit Secure Digital Host controller • Dual 10/100 Ethernet with L2 Switch • FlexCAN3
Memory Interfaces	• 8/16 bit DRAM Controller with support for LPDDR2/ DDR3 - Up to 800 MHz data rate • 8/16-bit NAND Flash controller with ECC • 8/16/32 bit External bus (Flexbus) • Dual QuadSPI supporting Execute-In-Place (XIP)

Table continues on the next page...

Table 2-1. Feature Set (continued)

Graphics Display and Video	• Dual Display Control Unit (DCU) with support for color TFT displays up to SVGA • Segment LCD (3V Glass only) configurable as 40x4, 38x6, and 36x8 • Video Interface Unit (VIU) for camera input • Run Length Endoded (RLE) Decoder • Composite Video Decoder supporting PAL and NTSC
Analog	• 12-bit SAR ADC • 12-bit DAC
Audio	• Synchronous Audio Interfaces (SAI) supporting I2S, AC97 and Codec/DSP interfaces • Enhanced Serial Audio Interface (ESAI) • Sony/Philips Digital Interface (SPDIF), Rx and Tx • Asynchronous Sample Rate Converter (ASRC)
Human-Machine Interface (HMI)	• GPIO pins with interrupt support, DMA request capability, digital glitch filter. • Hysteresis and configurable pull up/down device on all input pins • Configurable slew rate and drive strength on all output pins
On-Chip RAM/ROM	• 512 KB On-Chip SRAM with ECC • 1 MB On-Chip Graphics SRAM without ECC • 96KB On-Chip ROM
Power Consumption	• Low power modes (LP/ULPRUN, STOP, LPSTOP1, LPSTOP2 and LPSTOP3)

2.3 Detailed Block Diagram

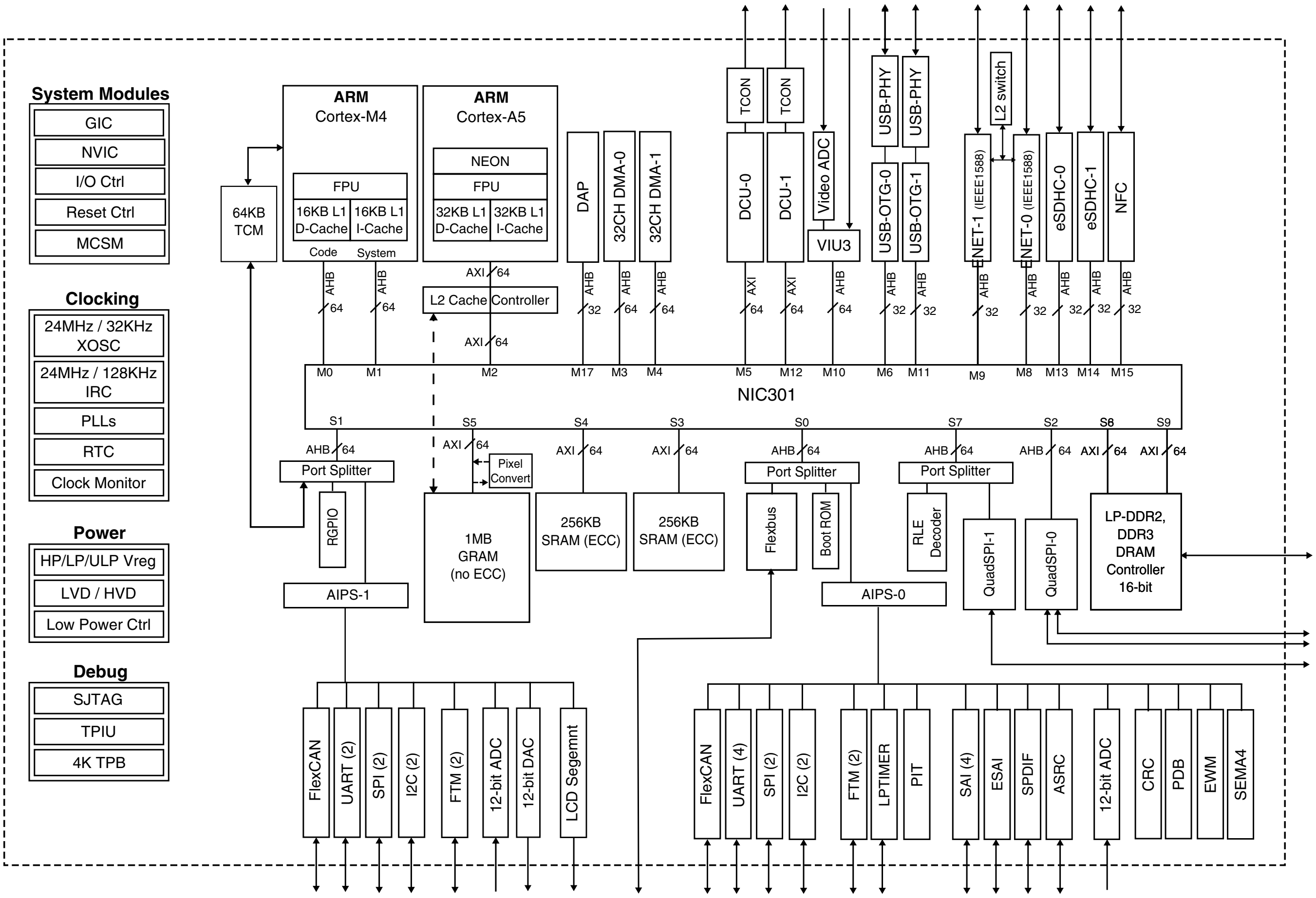


Figure 2-1. Detailed Block Diagram

NOTE

Throughout this manual, ARM® Cortex®-A5 is referred as Cortex-A5 or CA5. Also, ARM® Cortex®-M4 is referred as Cortex-M4 or CM4.

2.4 Device Configuration

The following table lists the superset configuration for each package within the device.

Table 2-2. Device Configuration

Device	VF3XX	VF5XX	VF6XX
General			
Cortex®-A5 Core Frequency	266 MHz	Up to 500 MHz	Up to 500 MHz
Cortex®-M4 Core Frequency	N/A	N/A	Up to 167 MHz
Package	176 LQFP	364 BGA	364 BGA

Table continues on the next page...

Table 2-2. Device Configuration (continued)

Device	VF3XX	VF5XX	VF6XX
Package Dimensions (mm^2)	24 x 24 Pitch (0.5 mm)	17 x 17 Pitch (0.8mm)	17 x 17 Pitch (0.8 mm)
Core, Platform and Debug			
DMA (with DMA Mux)	2 x 32 ch DMA Controllers 4 x 64 to 16 DMA Muxes	2 x 32 ch DMA Controllers 4 x 64 to 16 DMA Muxes	2 x 32 ch DMA Controllers 4 x 64 to 16 DMA Muxes
Trace Port	8 bit Output Trace	16 bit Output Trace	16 bit Output Trace
Security Subsystem			
Random number generator accelerator (RNG)	1, (NIST SP 800-90)	1, (NIST SP 800-90)	1, (NIST SP 800-90)
External Tamper inputs	2	6	6
On-Chip Memories			
On-chip RAM	1.5 MB	1.5 MB SRAM or 1 MB SRAM + 512 KB L2 Cache	1.5 MB SRAM or 1 MB SRAM + 512 KB L2 Cache
ECC on On-chip RAM (512K)	1	1	1
On-Chip ROM	96 KB	96 KB	96 KB
Memory Interfaces			
DRAM Controller (LPDDR2/ DDR3)	No	16 bit	16 bit
NAND Flash Controller	1 (8 bit)	1 (16 bit)	1 (16 bit)
Quad SPI	2	2	2
Timers/PWM			
FlexTimer (FTM0) channel pins	8 Ch	8 Ch	8 Ch
FlexTimer (FTM1) channel pins	2 Ch	2 Ch	2 Ch
FlexTimer (FTM2) channel pins	2 Ch	2 Ch	2 Ch
FlexTimer (FTM3) channel pins	None	8 Ch	8 Ch
IEEE® 1588 Timers	4 Ch	8 Ch	8 Ch
Periodic Interrupt Timers(PITs)	4 Ch	8 Ch	8 Ch
Low power timer (LPTMR)	1	1	1
Communication Interfaces			
10/100 ENET with IEEE®1588	2	2	2
eSDHC	1 (8 bit)	2 (8 bit and 4 bit)	2 (8 bit and 4 bit)
FlexCAN	2	2	2
USB 2.0 HS OTG Controller	1	2	2
UART (SCI)	4	6	6
DSPI (16-bit)	3	4	4
IIC	2	4	4

Table continues on the next page...

Table 2-2. Device Configuration (continued)

Device	VF3XX	VF5XX	VF6XX
Display and Video			
TFT Display Control Unit (DCU)	1	2	2
Segmented LCD (40x4)	1	0	0
Video Interface Unit (VIU)	1 ((digital input only)	1 (digital input only)	1 (with digital or analog video input)
RLE Decoder	1	1	1
Analog			
12 bit SAR ADC	2 (12 Ch)	2 (16 Ch)	2 (16 Ch)
Video ADC (Channels)	0	0	4
USB HS PHY (OTG)	1	2	2
12-bit DAC	2	2	2
Audio			
Asynchronous Sample Rate converter(ASRC)	1	1	1
SAI	3	4	4
ESAI	1	1	1
SDPIF	1	1	1
Human Machine Interface			
Total GPIO pins (with Interrupt capability)	Up to 115	Up to 135	Up to 135

2.5 Modules on the device

2.5.1 Clocks

The following clock modules are available on this device.

Table 2-3. Clock modules

Module	Description	Reference links to the related information [1]
Clock Controller Module (CCM)	The Clock Controller Module controls the following functions: • Uses the available clock sources to generate clock roots to various parts of the device • Uses programmable bits to control frequencies of the clock roots • Controls the low power mechanism • Provides control signals to Low Power Clock Gating module (LPCG) for gating clocks • Provides handshake with System Reset Controller (SRC) for reset performance • Provides handshake with Global Power Controller (GPC) for low power mode operations	• Clocking Overview to read about the overall clock distribution on this device. It provides an overview about the different clock sources their typical configuration, and module clocks on this device. • Clock Controller Module (CCM) to read about module features, programing model, Low Power Clock gating, and auxillary clocks on this device.
Analog components control interface (ANADIG)	ANADIG is collection of digital interfaces and controllers of analog components, which include control registers for controlling analog components like device PLLs, PFDs, and regulators.	• Clocking Overview. • ANADIG to read about the registers that control the PLLs, PFDs, and regulators on this device.
Slow Clock Source Controller (SCSC)	Controls the module configures the Slow internal RC oscillator 128 KHz (SIRC) and Slow external crystal oscillator 32 KHz (SXOSC) on the device.	• Clocking Overview. • Introduction Slow Clock Source Controller (SCSC)
Clock Monitor Unit (CMU)	Measures the frequency of clock sources	• Clocking Overview. • Clock Monitor Unit (CMU) to read about module features, programming model, and signals. • CMU Chip Signals section to view the mapping between CMU module signals and the chip-level signals.

1. It is recommended to read the information in the order below.

2.5.2 Platform Modules

The following table lists all platform modules available on this device.

NOTE

Shaded modules are ARM modules. For the detailed description about them, refer to the ARM website.

Table 2-4. Platform modules

Module	Description	Reference

ARM Cortex-A5	The Cortex-A5 processor is a high-performance, low-power ARM macrocell with an L1 cache subsystem that provides full virtual memory capabilities. It supports:
	• ARMv7-A instruction set architecture including
	  • Includes Vector Floating-Point v4 (VFPv4) architecture
	  • NEON Media Processing Engine (MPE)
	  • TrustZone security
	• 8-stage single issue pipeline implementation operating at 266.6, 400-500 MHz speeds
	  • 1.57 DMIPS per MHz integer performance
	  • 4-stage load/store pipeline
	  • 5-stage FPU/MPE pipeline
	• 64-bit AXI System Bus Interface supporting multiple outstanding transactions
	• Processor-local Memories
	  • 2 way set-associative 32 KB Instruction Cache with 32 byte line size
	  • 4 way set-associative 32 KB Data Cache with 32 byte line size
	  • Standard Cortex-A5 Memory Management Unit 64-bit AXI System Bus Interface supporting multiple outstanding transactions
		For ARM Cortex-A5 processor documentation, refer to Cortex-A5 MPCore r0p1 Technical Reference Manual, DDI0434B at http://www.arm.com

CoreLink™ Level 2 Cache Controller	The addition of an on-chip secondary cache, also referred to as a Level 2 or L2 cache, is a recognized method of improving the performance of ARM-based systems when significant memory traffic is generated by the processor. By definition a secondary cache assumes the presence of a Level 1 or primary cache, closely coupled or internal to the processor.	For ARM Cache Controller documentation, refer to DDI0246F_l2c310_r3p2_trm at http://www.arm.com

Table continues on the next page...

Table 2-4. Platform modules (continued)

Module	Description	Reference

ARM Cortex-M4 Core	The Cortex-M4 processor core brings next-generation capabilities to its predecessor, the Cortex-M3. Its new capabilities provide backward compatibility with the Cortex-M3, while adding important features.
	• Supports ARMv7-M instruction set architecture
	  • Includes the single precision FPU
	• 3-stage single issue pipeline implementation
	  • 1.25 DMIPS per MHz integer performance
	• Processor-local Memories
	  • All local memories operate at core frequency and provide 0 wait state response on “hits”
	  • 64 KB of Tightly-Coupled Memory split equally between TCM{Lower, Upper}
	  • 2 way set-associative 16KB CodeCache
	  • 2 way set-associative 16 KB SystemCache
	• Modified Harvard 64-bit AHB System Bus Interface + 64-bit AHB backdoor port to TCM
		For ARM Cortex-M4 processor documentation, refer to Cortex-M4 User Guide Reference Material, DUI0553A at http://www.arm.com

Local Memory Controller	The Local Memory Controller provides the ARM Cortex-M4 processor with tightly coupled processor-local memories and bus paths to all slave memory spaces. The local memory controller includes four memory controllers and their attached memories: • SRAM lower (SRAM_L) controller via the PC bus • SRAM upper (SRAM_U) controller via the PS bus • Cache memory controller via the PC bus • Cache memory controller via the PS bus	

Table continues on the next page...

Table 2-4. Platform modules (continued)

Module	Description	Reference
CoreLink™ System Bus Interconnect (NIC301)	The CoreLink™ Network Interconnect (NIC301) is a 2nd generation highly configurable IP component that enables the creation of a complete high performance, optimized AMBA-compliant network infrastructure. • 1-128 AXI or AHB-Lite slave interfaces for bus master connections • 1-64 master interfaces that can be AXI, AHB-Lite, APB2, or APB3 for bus slave connections • Single-cycle arbitration • Full pipelining to prevent master stalls • Programmable control for FIFO transaction release • Multiple switch networks • AXI or AHB-Lite masters and slaves • Non-contiguous APB slave address map for a single master interface • Independent widths of user-defined sideband signals for each channel • Global Programmers View (GPV) for the entire infrastructure, configurable for customizing the memory mapped visibility • Highly flexible timing closure options	For NIC301 documentation, refer to DDI0397H_corelink_network_interconnect_nic301_r2p2_trm.pdf at http://www.arm.com

ARM Generic Interrupt Controller (GIC)	Generic Interrupt Controller (GIC) is a centralized resource for supporting and managing interrupts in a system that includes at least one processor. It provides:
	• registers for managing interrupt sources, interrupt behavior, and interrupt routing to one or more processors
	• support for the following:
	    • the ARM architecture Security Extensions
	    • the ARM architecture Virtualization Extensions
	    • enabling, disabling, and generating processor interrupts from hardware (peripheral) interrupt sources
	    • Software-generated Interrupts (SGIs)
	    • interrupt masking and prioritization
	    • uniprocessor and multiprocessor environments
	    • wakeup events in power-management environments.
		For ARM GIC documentation, refer to IHI0048B_gic_architecture_specification.pdf at http://www.arm.com

Table continues on the next page...

Table 2-4. Platform modules (continued)

Module	Description	Reference
Debug Interfaces	The device debug and trace is based on ARM CoreSight^{TM} architecture supplemented with the Secured JTAG controller (SJC) to allow security features. Debug interfaces supported are: • IEEE 1149.1 System JTAG Controller (SJC) that provides the security authentication for debug access to the chip. • IEEE 1149.7 JTAG. Also known as compact JTAG (cJTAG). • ARM Serial Wire Debug (SWD) • Support for Secured and non secured invasive/ non invasive debug to allow further granularity in debug accesses. • Support for field return parts to open access for debug and test to allow failure analysis. • Cross Trigger supported between the two cores as recommended by ARM. • Program trace support • Data trace supported by the Cortex-A5	For ARM debug documentation, refer to ARM Debug Interface - IHI0031A_ARM_debug_interface_v5.pdf at http://www.arm.com
eDMA	• 32 channels support independent • 8-, 16-, or 32-bit single value or block transfers • Supports variable sized queues and circular queues • Source and destination address registers are independently configured to postincrement or remain constant • Each transfer is initiated by a peripheral, CPU, periodic timer interrupt or eDMA channel request • Each DMA channel can optionally send an interrupt request to the CPU on completion of a single value or block transfer • DMA transfers possible between memories, General Purpose I/Os (GPIOs) and Slave Peripherals that support DMA • Programmable DMA Channel Mux allows assignment of any DMA source to any available DMA channel with up to a total of 64 potential request sources	Refer to the Direct Memory Access Controller (eDMA) chapter of this manual.
Peripheral Bridge (AIPS-Lite)	• Supports up to 160 peripherals and two global external peripheral spaces • Supports 8-, 16-, and 32-bit width peripheral slots • Supports a pair of 32-bit transactions for selected 64-bit memory accesses • Each independently configurable peripheral includes a clock enable, which allows peripherals to operate at any speed less than the system clock rate.	Refer to the AIPS-lite chapter of this manual.

Table continues on the next page...

Table 2-4. Platform modules (continued)

Module	Description	Reference

Semaphores (SEMA4)	The Semaphores module implements hardware-enforced semaphores as an IPS-mapped slave peripheral device. The feature set includes:
	• Support for 16 hardware-enforced gates in a dual-processor configuration
	  • Each hardware gate appears as a 3-state, 2-bit state machine, with all 16 gates mapped as a byte-size array
	  • Optional interrupt notification after a failed lock write provides a mechanism to indicate when the gate is unlocked
	  • Secure reset mechanisms are supported to clear the contents of individual gates or notification logic, as well as a clear_all capability
	• Memory-mapped IPS slave peripheral platform module
	  • Interface to the IPS bus for programming-model accesses
	  • Two outputs (one per processor) for interrupt notification of failed lock writes
		Refer to the IPS_Semaphore chapter of this manual.

2.5.3 System Modules

The following system modules are available on this device.

Table 2-5. System modules

Module	Description	Reference links to the chip related information
Power management controller (PMC)	The PMC provides the user with multiple power options. Ten different modes are supported that allow the user to optimize power consumption for the level of functionality needed. Includes power-on-reset (POR) and integrated low voltage detect (LVD) with reset (brownout) capability.	
Peripheral bridges	The peripheral bridge converts the crossbar switch interface to an interface to access a majority of peripherals on the device.	
DMA multiplexer (DMAMUX)	The DMA multiplexer selects from many DMA requests down to 16 for the DMA controller. There are 2 DMA multiplexers associated with each 32-channel DMA.	DMAMUX Request Sources
Direct memory access (DMA) controller	The DMA controller provides programmable channels with transfer control descriptors for data movement via dual-address transfers for 8-, 16-, 32- and 128-bit data values.	

Table continues on the next page...

Table 2-5. System modules (continued)

Module	Description	Reference links to the chip related information
External watchdog monitor (EWM)	The EWM is a redundant mechanism to the software watchdog module that monitors both internal and external system operation for fail conditions.	
Software watchdog (WDOG)	The WDOG monitors internal system operation and forces a reset in case of failure. It operates on SXOSC 32 KHz clock with a programmable refresh window to detect deviations in program flow or system frequency.	

2.5.4 Memories and Memory Interfaces

The following memories and memory interfaces are available on this device.

Table 2-6. Memories and memory interfaces

Module	Description	Reference links to chip related information

On-chip memory	• The device Includes:
	  • 512 KB On-Chip SRAM with error-correcting code (ECC)
	  • 1 MB On-Chip Graphics SRAM without ECC
	  • 96 KB On-Chip ROM	

Programmable Cyclic Redundancy Check (CRC32)	The module generates 16/32-bit CRC code for error detection. The CRC module provides a programmable polynomial, SEED, and other parameters required to implement a 16-bit or 32-bit CRC standard. The 16/32-bit code is calculated for 8/16/32-bit data at a time. The data width and transpose features are selectable by a parameter.	

Table continues on the next page...

Table 2-6. Memories and memory interfaces (continued)

Module	Description	Reference links to chip related information
DRAM Memory Controller (DDRMC)	DDRMC is a complete embedded memory controller that interfaces to a PHY. The features of this Memory Controller include: • Supports interfacing to LPDDR2 and DDR3 memory types. It supports 8-bit and 16-bit memory interface. • Fully pipelined command, read and write data interfaces to the memory controller. • Advanced bank look-ahead features for high memory throughput. • Front-end interface to 2 standard AXI ports. A programmable register interface to control memory parameters and protocols including auto pre-charge. • Full initialization of memory on memory controller reset. • ECC functionality with single bit and double bit error reporting and automatic correction of single bit error events. 10-bit memory interface is required for ECC (8-bit user data + 2-bit ECC). • ECC functionality with single bit and double bit error reporting and automatic correction of single bit error events. • Clock frequencies from 100 MHz to 400 MHz supported. • Back-end interface to a PHY. • Integrates support for DDR pad calibration logic - both software and hardware auto modes	DDR maximum address space
FlexBus	External bus interface with multiple independent, user-programmable chip-select signals that can interface with external SRAM, PROM, EPROM, EEPROM, flash, and other peripherals via 8-, 16- and 32-bit port sizes. Configurations include multiplexed or non-multiplexed address and data buses using 8-bit, 16-bit, 32-bit, and 16-byte line-sized transfers. Maximum frequency that FlexBus supports is 57 MHz	• FlexBus signal multiplexing • FlexBus external signal • FlexBus security • Instantiation Information • FlexBus Chip Select Control Register (CSCR0) Reset Value • Bus Timeout
NAND flash controller	8-bit and 16-bit NAND flash interface with 32-bit ECC error correction. • Supports all NAND Flash products regardless of density/organization (with page size of 512 K+16B/2 K+64B/4 K +128B/4 K+218B/8 K) • Two Configurable DMA channels • Maximum serial clock frequency 80 MHz	Instantiation

Table continues on the next page...

Table 2-6. Memories and memory interfaces (continued)

Module	Description	Reference links to chip related information
QuadSPI	Interface for up to two external quad serial flash memories for code / data storage and code execution • Supports industry standard single, dual and quad mode serial flashes • Supports Double Data Rate (DDR) serial flash for high performance • Maximum serial clock frequency 80 MHz • Dual controller architecture enables simultaneous access to two external flashes resulting peak read bandwidth of 160 Mbytes/s • Flexible buffering scheme, multi-master, prioritized access	• QuadSPI Instances • QuadSPI Memory Interface • QuadSPI Buffer • Booting from QuadSPI

2.5.5 Audio modules

The following audio modules are available on this device:

Table 2-7. Audio modules

Module	Description	Reference links to chip related information
Enhanced Serial Audio Interface (ESAI)	Provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, Sony/Phillips Digital Interface (SPDIF) transceivers, and other DSPs. The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator.	• Enhanced Serial Audio Interface (ESAI) • ESAI Bus Interface and FIFO (ESAI_BIFIFO)
Sony/Philips Digital Interface (SPDIF)	The SPDIF is composed of two parts: • SPDIF Receiver: The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in the SPDIF Rx left and right FIFOs. The Channel Status and User Bits are also extracted from each frame and placed in the corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter. • SPDIF Transmitter. For the SPDIF transmitter, the audio data is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers. The Channel Status bits are also provided via the corresponding registers.	

Table continues on the next page...

Table 2-7. Audio modules (continued)

Module	Description	Reference links to chip related information
Asynchronous Sample Rate Converter (ASRC)	Converts the sampling rate of a signal associated to an input clock into a signal associated to a different output clock.The ASRC supports concurrent sample rate conversion of up to 10 channels of about -120dB THD+N. The ASRC supports up to 3 sampling rate pairs. The ASRC is hard-coded implemented, as a co-processor, with minimal CPU intervention.	
Synchronous Audio Interface (SAI)	Implements supports full-duplex serial interfaces with frame synchronization such as I2S, AC97, and CODEC/ DSP interfaces.	• Synchronous Audio Interface (SAI) • SAI3 register details • Simultaneous SAI DMA requests • SAI transmitter and receiver options for MCLK selection • SAI in Stop mode

2.5.6 Timer modules

The following timer modules are available on this device:

Table 2-8. Timer modules

Module	Description	Reference links to chip related information
Programmable delay block (PDB)	• 16-bit resolution • 3-bit prescaler • Positive transition of trigger event signal initiates the counter • Supports two triggered delay output signals, each with an independently-controlled delay from the trigger event • Outputs can be OR'd together to schedule two conversions from one input trigger event and can schedule precise edge placement for a pulsed output. This feature is used to generate the control signal for the CMP windowing feature and output to a package pin if needed for applications, such as critical conductive mode power factor correction. • Continuous-pulse output or single-shot mode supported, each output is independently enabled, with possible trigger events • Supports bypass mode • Supports DMA	• PDB Instantiation • PDB Module Interconnections • DMA support on PDB • PDB in Low-Power modes • PDB implementation with ADC

Table continues on the next page...

Table 2-8. Timer modules (continued)

Module	Description	Reference links to chip related information
Flexible timer modules (FTM)	• Selectable FTM source clock, programmable prescaler • 16-bit counter supporting free-running or initial/final value, and counting is up or up-down • Input capture, output compare, and edge-aligned and center-aligned PWM modes • Operation of FTM channels as pairs with equal outputs, pairs with complimentary outputs, or independent channels with independent outputs • Deadtime insertion is available for each complementary pair • Generation of hardware triggers • Software control of PWM outputs • Up to 4 fault inputs for global fault control • Configurable channel polarity • Programmable interrupt on input capture, reference compare, overflowed counter, or detected fault condition • Quadrature decoder with input filters, relative position counting, and interrupt on position count or capture of position count on external event • DMA support for FTM events	• FTM Instantiation • FTM output triggers for other modules • FTM Global Time Base • FTM Hardware Triggers • FTM Fault Detection Inputs
Periodic interrupt timer (PIT)	• 8 channels • Interrupt timers for triggering ADC conversions • 32-bit counter resolution • Clocked at system clock frequency • 4 channels connected to DMA	• PIT Instantiations • PIT/DMA Periodic Trigger Assignments
Low-power timer (LPTimer)	• Selectable clock for prescaler/glitch filter of 1 kHz (internal LPO), 32.768 kHz (external crystal), or internal reference clock • Configurable Glitch Filter or Prescaler with 16-bit counter • 16-bit time or pulse counter with compare • Interrupt generated on Timer Compare • Hardware trigger generated on Timer Compare	LPTMR prescaler/glitch filter clocking options
Real-time clock (RTC)	• Independent power supply, POR, and 32 kHz Crystal Oscillator • 32-bit seconds counter with 32-bit Alarm • 16-bit Prescaler with compensation that can correct errors between 0.12 ppm and 3906 ppm	

2.5.7 Communication interfaces

The following communication interfaces are available on this device:

Table 2-9. Communication modules

Module	Description	Reference links to the chip related information

Ethernet MAC with IEEE 1588 capability (ENET)	Ethernet Subsystem on Vybrid include the following blocks:
	• Dual 10/100 Ethernet MAC (MAC-NET From MTIP)
	  • Hardware support for IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems IEEE 1588
	  • Reduced media independent interface (RMII) support
	  • Interfaces with Unified DMA
	  • Supports wake-up from low power mode through magic packets
	  • Multiple clock source options for time-stamping clock
	• 10/100 L2 Ethernet Switch (From MTIP)
	  • 3-Port Switch
	  • Supports two MAC-NETs
	  • Supports 64-bit Atlantic/FIFO ports and IEEE 1588 support
	  • Fast cut-through mode
	  • QoS with 8-queues per port and port mirroring
	  • Level 3 IP snooping
	• Dual Unified DMA
	  • On-chip transmit and receive FIFOs
		• Instantiation Information
		• MII and RMII configuration
		• IEEE 1588 Timers
		• Ethernet Operation in Low Power Modes
		• Ethernet Subsystem Interrupts
		• Ethernet switch register reset values)

Universal Serial Bus Controller (USB)	USB 2.0 compliant module with support for host, device, and On-The-Go modes. Includes an on-chip transceiver for full and low speeds. The registers and data structures are based on the Enhanced Host Controller Interface Specification for Universal Serial Bus (EHCI) from Intel Corporation. **NOTE:** OTG controller should be treated as Dual role controller that allows the controller to act as either a Host or a device with no support for HNP/SRP.	• USB Configuration and Options • SOF for USB Audio • OverCurrent and VBUS Connection
Controller Area Network (CAN)	Supports the full implementation of the CAN Specification Version 2.0, Part B	FlexCAN Instantiation
Serial Peripheral Interface (SPI)	Synchronous serial bus for communication to an external device	• SPI Instantiation • Number of PCS
Inter-Integrated Circuit (I2C)	Allows communication between a number of devices. Also supports the System Management Bus (SMBus) Specification, version 2.	Instantiation Information
Universal asynchronous receiver/transmitters (UART)	Asynchronous serial bus communication interface with programmable 8- or 9-bit data format and support of CEA709.1-B (LON), ISO 7816 smart card interface	• UART configuration information • UART wakeup • UART interrupts • UART Instances Register Difference

Table continues on the next page...

Table 2-9. Communication modules (continued)

Module	Description	Reference links to the chip related information
Secure Digital host controller (SDHC)	Interface between the host system and the SD, SDIO, MMC, or CE-ATA cards. The SDHC acts as a bridge, passing host bus transactions to the cards by sending commands and performing data accesses to/from the cards. It handles the SD, SDIO, MMC, and CE-ATA protocols at the transmission level.	• SD bus pullup/pulldown constraints • SDHC Wakeup • SDHC Software Guidelines

2.5.8 Graphics Modules

The following core modules are available on this device.

Table 2-10. Graphics/Display Modules

Module	Description	Reference links to chip related information
Display Controller Unit (DCU4)	It is a system master that fetches graphics stored in internal or external memory and displays them on a TFT LCD panel. It supports: • Full RGB888 output to TFT LCD panel • 64 graphics layers, a default background color layer and a cursor layer with integrated blinking option • Blending of each pixel using up to 6 source layers dependent on size of panel • Programmable panel size up to XGA (1024x768) • Gamma correction with 8-bit resolution on each color component • Safety mode for tagging pixels on highest priority layers • Dedicated memory blocks to store a cursor and Color Look Up Tables (CLUTs) • Temporal Dithering	DCU Instantiation

Table continues on the next page...

Table 2-10. Graphics/Display Modules (continued)

Module	Description	Reference links to chip related information
RLE	Decodes data that has been compressed using a Run Length Encoding (RLE) scheme. It has input and output FIFO buffers directly connected to the crossbar switch and requires the CPU or DMA to push in the encoded data and then extract the decoded result. The module configuration is optimized for decoding data stored in a two-dimensional image format but can also be used to extract data stored as a linear array. • 32 channels support independent • 8/16/32-bit single value or block transfers • Supports variable sized queues and circular queues • Source and destination address registers are independently configured to postincrement or remain constant • Each transfer is initiated by a peripheral, CPU, periodic timer interrupt or eDMA channel request • Each DMA channel can optionally send an interrupt request to the CPU on completion of a single value or block transfer • DMA transfers possible between system memories, General Purpose I/Os (GPIOs) and Slave Peripherals that support DMA • Programmable DMA Channel Mux allows assignment of any DMA source to any available DMA channel with up to a total of 64 potential request sources	RLE Instantiation
TCON	The Timing Controller module (TCON) provides an alternative interface for the DCU that provides RGB data and timing signals for "raw" TFT panels which have no embedded TCON. • Flexible timing generation unit supporting 12 timing signal channels • Supports bit mapping of 8-bit or 6-bit color depth • Blanking of RGB data during inactive period (driven to all "0" or all "1")	TCON Instantiation

2.5.9 Analog modules

The following analog modules are available on this device:

Table 2-11. Analog modules

Module	Description	Reference links to chip related information
12-bit analog-to-digital converter (ADC)	It is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.	• ADC Instantiation • Voltage reference selection (REFSEL settings) • DMA Support on ADC • ADC Channel Assignments • ADC interconnections
Video ADC	The VideoADC module comprises an analog front end and video decoder block allowing up to 4 composite video sources to be connected. The Video ADC accepts NTSC and PAL format composite video, digitises, filters and decode this video and outputs a digital video stream to the VIU module for further processing.	Instantiation
Temperature Voltage Monitor	It contains the voltage and temperature monitor circuits. These circuits are put into a separate power domain. Its features are: • Low power designs to support the coin battery • Trimmable Voltage and Temperature detector thresholds	
Power Management Unit (PMU)	The PMU of the device includes: • High power or main regulator (HPREG) • Voltage reference for HPREG • Low power regulator (LPREG) • Ultra low power regulator (ULPREG) • Voltage reference for LPREG and ULPREG • Low voltage detector for 3.3V supply • Separate Low voltage detectors for HPREG ,LPREG ,ULPREG output voltages · Power on Reset(POR) • Power up sequencing and testing • N-well bias circuit	

Chapter 3
Memory Map

3.1 System memory map

The following table shows the high-level device memory map.

CM4 Address Range [Start Addr - End Addr]	Size [MB]	CM4 Alias	System Address (A5) [Start Addr - End Addr]	Region Description
0x0000_0000 -0x007f_ffff	8	0x0000_0000	0x0000_0000-0x007f_ff ff	Boot ROM
0x0080_0000-0x0fff_ffff	248	0x8080_0000	Reserved	CM4 DDR code alias
0x1000_0000-0x17ff_fff f	128	0x2000_0000	Reserved	CM4 QuadSPI0 code alias
0x1800_0000-0x1eff_fff f	112	0x3000_0000	Reserved	CM4 FlexBus code alias
0x1f00_0000-0x1f7f_ffff	8	0x3f00_0000	Reserved	CM4 OCRAM code alias
0x1f80_0000-0x1fff_ffff	8	N/A	0x1f80_0000-0x1fff_ffff	CM4 TCML (code)
0x2000_0000-0x2fff_ffff	256	0x2000_0000	0x2000_0000-0x2fff_ffff	QuadSPI0 Memory
0x3000_0000-0x3eff_fff f	240	0x3000_0000	0x3000_0000-0x3eff_fff f	FlexBus
0x3f00_0000- 0x3F03_FFFF	0.25	0x3f00_0000	0x3f00_0000-0x3F03_F FFF	OCRAM - SysRAM0
0x3F04_0000 - 0x3F07_FFFF	0.25	0x3F04_0000	0x3F04_0000 - 0x3F07_FFFF	OCRAM - sysRAM1
0x3F08_0000 - 0x3F3F_FFFF	3.50	N/A	Reserved	Reserved
0x3f40_0000-0x3f47_fff f	.5	0x3f40_0000	0x3f40_0000-0x3f47_fff f	OCRAM - gfxRAM
0x3f48_0000-0x3f7f_ffff	3	N/A	0x3f50_0000-0x3f7f_ffff	Reserved
0x3f80_0000-0x3fff_ffff	8	N/A	0x3f80_0000-0x3fff_ffff	CM4 TCMU (data)
0x4000_0000-0x4006_f fff	0.448	0x4000_0000	0x4000_0000-0x4006_f fff	IPS0

Table continues on the next page...

CM4 Address Range [Start Addr - End Addr]	Size [MB]	CM4 Alias	System Address (A5) [Start Addr - End Addr]	Region Description
0x4008_0000-0x400f_efff	0.5	0x4008_0000	0x4008_0000-0x400f_efff	IPS1
0x4010_0000-0x4fff_ffff	255	N/A	Reserved	Reserved
0x5000_0000-0x5fff_ffff	256		0x5000_0000-0x5fff_ffff	QuadSPI1 Memory
0x6000_0000 0x6fff_ffff	256	N/A	Reserved	Reserved
0x7000_0000 0x77ff_ffff	128	N/A	Reserved	Reserved
0x7800_0000 0x79ff_ffff	32	0x7800_0000	0x7800_0000 - 0x79ff_ffff	RLE
0x7a00_0000 0x7bff_ffff	32	0x7a00_0000	0x7a00_0000 - 0x7bff_ffff	QuadSPI1 Rx buffer
0x7c00_0000 0x7dff_ffff	32	0x7c00_0000	0x7c00_0000 - 0x7dff_ffff	QuadSPI0 Rx buffer
0x7e00_0000 0x7e7f_ffff	8	0x7e00_0000	0x7e00_0000 - 0x7e7f_ffff	gfxRAM- RGB565 view
0x7e80_0000 0x7eff_ffff	8	0x7e80_0000	0x7e80_0000 - 0x7eff_ffff	gfxRAM-ARGB1555 view
0x7f00_0000 0x7f7f_ffff	8	0x7f00_0000	0x7f00_0000 - 0x7f7f_ffff	gfxRAM-ARGB4444 view
0x7f80_0000 0x7fff_ffff	8	N/A	Reserved	Reserved
0x8000_0000-0xdfff_ffff	1536		0x8000_0000-0xdfff_ffff	DDR
0xe000_0000-0xffff_ffff	512	N/A	Reserved	CM4 Private Peripheral Bus (PPB)
	512	N/A	0xe000_0000-0xffff_ffff	DDR (A5 only)

NOTE

16 KB SRAM on the device occupies upper part of memory assigned to SRAM (0x4007_c000- 0x4007_FFFF)

NOTE

64 KB/16 KB SRAM, which is retained in LPSTOP modes occupies the lower part of OCRAM-SysRAM0

NOTE

For OCRAM-gfxRAM region (0x3f40_0000-0x3f47_ffff), if the "L2 Cache DISABLE " eFuse is blown (L2_CACHE_DISABLE = 1), the entire 1 MB space based at 0x3f40_0000 is allocated to gfxRAM. If L2_CACHE_DISABLE = 0, the first 512 KB (0x3f40_0000 - 0x3f47_ffff) is allocated to gfxRAM and the upper 512 KB is allocated to the L2 Cache data array.

NOTE

If there is an access to reserved/illegal memory space, then the system may hang, requiring a reset to recover. There is no time-out mechanism to recover from this scenario.

3.2 Peripheral Bridge 0 (AIPS-Lite 0) Memory Map

Table 3-2. Peripheral bridge 0 slot assignments

System 32-bit base address	Slot number	Module
On-platform		
0x4000_0000	0	
0x4000_1000	1	MSCM (CPU Configuration, INTR, OCRAM)
0x4000_2000	2	CA5-SCU+GIC CPU Interface registers[1]
0x4000_3000	3	CA5-INTD GIC Distributor registers[2]
0x4000_4000	4	
0x4000_5000	5	
0x4000_6000	6	CA5-L2C (CA5 L2 Cache Controller)
0x4000_7000	7	
0x4000_8000	8	NIC0 (Network Interconnect)
0x4000_9000	9	NIC1
0x4000_A000	10	NIC2
0x4000_B000	11	NIC3
0x4000_C000	12	NIC4
0x4000_D000	13	NIC5
0x4000_E000	14	NIC6
0x4000_F000	15	NIC7
0x4001_0000	16	
0x4001_1000	17	
0x4001_2000	18	
0x4001_3000	19	
0x4001_4000	20	
0x4001_5000	21	
0x4001_6000	22	
0x4001_7000	23	
0x4001_8000	24	DMA0
0x4001_9000	25	DMA0_TCD
0x4001_A000	26	
0x4001_B000	27	
0x4001_C000	28	

Table continues on the next page...

Table 3-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4001_D000	29	SEMA4
0x4001_E000	30	FlexBus
0x4001_F000	31	
Off-platform		
0x4002_0000	32	FlexCAN0
0x4002_1000	33	FlexCAN0
0x4002_2000	34	FlexCAN0
0x4002_3000	35	FlexCAN0
0x4002_4000	36	DMA channel Mux0
0x4002_5000	37	DMA channel Mux1
0x4002_6000	38	
0x4002_7000	39	UART0
0x4002_8000	40	UART1
0x4002_9000	41	UART2
0x4002_A000	42	UART3
0x4002_B000	43	
0x4002_C000	44	SPI 0
0x4002_D000	45	SPI 1
0x4002_E000	46	
0x4002_F000	47	SAI0
0x4003_0000	48	SAI1
0x4003_1000	49	SAI2
0x4003_2000	50	SAI3
0x4003_3000	51	CRC
0x4003_4000	52	USBC0
0x4003_5000	53	
0x4003_6000	54	Programmable delay block (PDB)
0x4003_7000	55	Periodic interrupt timers (PIT)
0x4003_8000	56	FlexTimer (FTM) 0
0x4003_9000	57	FlexTimer (FTM) 1
0x4003_A000	58	—
0x4003_B000	59	Analog-to-digital converter (ADC) 0
0x4003_C000	60	
0x4003_D000	61	TCON0
0x4003_E000	62	WDOG-A5
0x4003_F000	63	WDOG-M4
0x4004_0000	64	Low-power timer (LPTMR)
0x4004_1000	65	
0x4004_2000	66	RLE

Table continues on the next page...

Table 3-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4004_3000	67	-
0x4004_4000	68	QuadSPI0
0x4004_5000	69	
0x4004_6000	70	
0x4004_7000	71	
0x4004_8000	72	IO MUX Controller
0x4004_9000	73	Port A multiplexing control
0x4004_A000	74	Port B multiplexing control
0x4004_B000	75	Port C multiplexing control
0x4004_C000	76	Port D multiplexing control
0x4004_D000	77	Port E multiplexing control
0x4004_E000	78	
0x4004_F000	79	
0x4005_0000	80	ANADIG
0x4005_1000	81	
0x4005_2000	82	Slow Clock Source Controller Module (SCSC)
0x4005_3000	83	
0x4005_4000	84	
0x4005_5000	85	
0x4005_6000	86	
0x4005_7000	87	
0x4005_8000	88	DCU0
0x4005_9000	89	DCU0
0x4005_A000	90	DCU0
0x4005_B000	91	DCU0
0x4005_C000	92	DCU0
0x4005_D000	93	DCU0
0x4005_E000	94	DCU0
0x4005_F000	95	DCU0
0x4006_0000	96	ASRC
0x4006_1000	97	SPDIF
0x4006_2000	98	ESAI
0x4006_3000	99	
0x4006_4000	100	
0x4006_5000	101	External watchdog (EWM)
0x4006_6000	102	I^2C 0
0x4006_7000	103	I^2C 1
0x4006_8000	104	
0x4006_9000	105	

Table continues on the next page...

Table 3-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4006_A000	106	Wake up Unit (WKUP)
0x4006_B000	107	Clock Control Module (CCM)
0x4006_C000	108	Global Power Controller (GPC)
0x4006_D000	109	Voltage Regulator-Digital (VREG)
0x4006_E000	110	System Reset Controller (SRC)
0x4006_F000	111	Clock Monitor Unit (CMU)

1. Refer to ARM GIC documentation for details of these registers.
2. Refer to ARM GIC documentation for details of these registers.

3.3 Peripheral Bridge 1 (AIPS-Lite 1) Memory Map

Table 3-3. Peripheral bridge 1 slot assignments

System 32-bit base address	Slot number	Module
On-platform		
0x4008_0000	0	
0x4008_1000	1	
0x4008_2000	2	
0x4008_3000	3	
0x4008_4000	4	
0x4008_5000	5	
0x4008_6000	6	
0x4008_7000	7	Debug Access Port (DAP)-RomTable
0x4008_8000	8	CA5-DBG (CA5 Debug)
0x4008_9000	9	CA5-PMU (CA5 Performance Monitoring Unit)
0x4008_A000	10	CA5-ETM
0x4008_B000	11	
0x4008_C000	12	CA5-RomTable
0x4008_D000	13	
0x4008_E000	14	CA5-CTI
0x4008_F000	15	
0x4009_0000	16	CA5-Instrumentation Trace Macrocell (ITM)
0x4009_1000	17	CA5-Embedded Trace Macrocell (ETB)
0x4009_2000	18	CA5-Funnel
0x4009_3000	19	Pltf-TCTL
0x4009_4000	20	Pltf-Trace Port Interface Unit (TPIU)

Table continues on the next page...

Table 3-3. Peripheral bridge 1 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4009_5000	21	Pltf-Funnel
0x4009_6000	22	Pltf-Serial Wire Output (SWO)
0x4009_7000	23	
0x4009_8000	24	DMA1
0x4009_9000	25	DMA1_TCD
0x4009_A000	26	
0x4009_B000	27	
0x4009_C000	28	
0x4009_D000	29	
0x4009_E000	30	
0x4009_F000	31	
Off-platform		
0x400A_0000	32	
0x400A_1000	33	DMA Channel Mux2
0x400A_2000	34	DMA Channel Mux3
0x400A_3000	35	
0x400A_4000	36	
0x400A_5000	37	OTP CTRL
0x400A_6000	38	
0x400A_7000	39	
0x400A_8000	40	
0x400A_9000	41	UART4
0x400A_A000	42	UART5
0x400A_B000	43	
0x400A_C000	44	SPI 2
0x400A_D000	45	SPI 3
0x400A_E000	46	DDRMC
0x400A_F000	47	
0x400B_0000	48	
0x400B_1000	49	SDHC0
0x400B_2000	50	SDHC1
0x400B_3000	51	
0x400B_4000	52	USBC1
0x400B_5000	53	
0x400B_6000	54	
0x400B_7000	55	
0x400B_8000	56	FlexTimer (FTM) 2
0x400B_9000	57	FlexTimer (FTM) 3
0x400B_A000	58	

Table continues on the next page...

Table 3-3. Peripheral bridge 1 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x400B_B000	59	Analog-to-digital converter (ADC) 1
0x400B_C000	60	
0x400B_D000	61	TCON1
0x400B_E000	62	Segment LCD
0x400B_F000	63	
0x400C_0000	64	
0x400C_1000	65	
0x400C_2000	66	
0x400C_3000	67	
0x400C_4000	68	QuadSPI1
0x400C_5000	69	
0x400C_6000	70	
0x400C_7000	71	Video ADC
0x400C_8000	72	Video Decoder
0x400C_9000	73	VIU3
0x400C_A000	74	
0x400C_B000	75	
0x400C_C000	76	12-bit digital-to-analog converter (DAC) 0
0x400C_D000	77	12-bit digital-to-analog converter (DAC) 1
0x400C_E000	78	
0x400D_0000	80	Ethernet MAC0 and IEEE 1588 timers
0x400D_1000	81	Ethernet MAC1 and IEEE 1588 timers
0x400D_2000	82	
0x400D_3000	83	
0x400D_4000	84	FlexCAN1
0x400D_5000	85	FlexCAN1
0x400D_6000	86	FlexCAN1
0x400D_7000	87	FlexCAN1
0x400D_8000	88	DCU1
0x400D_9000	89	DCU1
0x400D_A000	90	DCU1
0x400D_B000	91	DCU1
0x400D_C000	92	DCU1
0x400D_D000	93	DCU1
0x400D_E000	94	DCU1
0x400D_F000	95	DCU1
0x400E_0000	96	Nand Flash Controller (NFC)
0x400E_1000	97	Nand Flash Controller (NFC)
0x400E_2000	98	Nand Flash Controller (NFC)

Table continues on the next page...

Table 3-3. Peripheral bridge 1 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x400E_3000	99	Nand Flash Controller (NFC)
0x400E_4000	100	
0x400E_5000	101	
0x400E_6000	102	I2C2
0x400E_7000	103	I2C3
0x400E_8000	104	Ethernet L2 Switch
0x400E_9000	105	Ethernet L2 Switch
0x400E_A000	106	Ethernet L2 Switch
0x400E_B000	107	Ethernet L2 Switch
0x400E_C000	108	Ethernet L2 Switch Look-up table
0x400E_D000	109	Ethernet L2 Switch Look-up table
0x400E_E000	110	Ethernet L2 Switch Look-up table
0x400E_F000	111	Ethernet L2 Switch Look-up table
0x400F_0000	112	
0x400F_1000	113	
0x400F_2000	114	
0x400F_3000	115	
0x400F_4000	116	
0x400F_5000	117	
0x400F_6000	118	
0x400F_7000	119	
0x400F_8000	120	
0x400F_9000	121	
0x400F_A000	122	
0x400F_B000	123	
0x400F_C000	124	
0x400F_D000	125	
0x400F_E000	126	
0x400F_F000	Not an AIPS-Lite slot. The 32-bit general purpose input/output module that shares the crossbar switch slave port with the AIPS-Lite is accessed at this address.	

NOTE

Access to non-implemented registers in following modules may not guarantee the device behavior:

- UART
- MLB
- NFC
- IOMUX
- ASRC

- SPDIF
- VIU
- TCON
- USB
- OCOTP
- ANADIG
- SDHC

3.4 Private Peripheral Bus (PPB) memory map

The PPB is part of the defined ARM bus architecture and provides access to select processor-local modules. These resources are only accessible from the core; other system masters do not have access to them.

Table 3-4. PPB memory map

System 32-bit Address Range	Resource
0xE000_0000–0xE000_0FFF	Instrumentation Trace Macrocell (ITM)
0xE000_1000–0xE000_1FFF	Data Watchpoint and Trace (DWT)
0xE000_2000–0xE000_2FFF	Flash Patch and Breakpoint (FPB)
0xE000_3000–0xE000_DFFF	Reserved
0xE000_E000–0xE000_EFFF	System Control Space (SCS) (for NVIC and FPU)
0xE000_F000–0xE004_0FFF	Reserved
0xE004_1000–0xE004_1FFF	Embedded Trace Macrocell (ETM)
0xE004_2000–0xE004_2FFF	Embedded Trace Buffer (ETB)
0xE004_3000–0xE004_3FFF	Embedded Trace Funnel
0xE004_4000–0xE004_4FFF	Cross Trigger Interface (CTI)
0xE005_0000–0xE007_4FFF	Reserved
0xE008_0000–0xE008_0FFF	Miscellaneous Control Module (MCM)
0xE008_1000–0xE008_1FFF	Reserved
0xE008_2000–0xE008_2FFF	Cache Controller
0xE008_3000–0xE00F_EFFF	Reserved
0xE00F_F000–0xE00F_FFFF	ROM Table - allows auto-detection of debug components

Chapter 4
Signal Multiplexing

4.1 Pinouts

The following table shows the signals available on each pin and the locations of these pins on the devices supported by this document. The IOMUX Controller (IOMUXC) Module is responsible for selecting which ALT functionality is available on each pin.

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
Y2	—	ADC0SE8	N/ A		ADC0_SE8							
W2	—	ADC0SE9			ADC0_SE9							
W3	—	ADC1SE8			ADC1_SE8							
Y3	—	ADC1SE9			ADC1_SE9							
W1	41	VREFH_ADC			VREFH_ADC							
U3	40	VREFL_ADC			VREFL_ADC							
V1	38	VDDA33_ADC			VDDA33_ADC							
V2	39	VSSA33_ADC			VSSA33_ADC							
U1	36	DACO0			DACO0							
U2	37	DACO1			DACO1							
Y4	—	VADCSE0			VADCSE0							
U4	—	VADCSE1			VADCSE1							
W4	—	VADCSE2			VADCSE2							
V5	—	VADCSE3			VADCSE3							
V3	—	VDDA33_AFE			VDDA33_AFE							
V4	—	VSSA33_AFE			VSSA33_AFE							
T5	—	VDD12_AFE			VDD12_AFE							
R5	—	VSS12_AFE			VSS12_AFE							
U5	—	VADC_AFE_ BANDGAP			VADC_AFE_ BANDGAP							
Y13	73	EXTAL			EXTAL							
W13	72	XTAL			XTAL							
Y12	70	EXTAL32			EXTAL32							

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
W12	71	XTAL32			XTAL32							
T4	35	RESETB/ RESET_OUT	RESETB/ RESET_OUT		RESETB/ RESET_OUT							
N5	19	PTA6		PTA6	RMII_CLKOUT	RMII_CLKIN/ MII0_TXCLK		DCU1_ TCON11			DCU1_R2	
T3	34	TEST			TEST							
T1	30	Ext_POR			TEST2							
V12	69	DECAP_V11_ LDO_OUT			DECAP_V11_ LDO_OUT							
T11	65	DECAP_V25_ LDO_OUT			DECAP_V25_ LDO_OUT							
T2	33	BCTRL			BCTRL							
P5	31	VDDREG			VDDREG							
T12	68	VDD33_ LDOIN			VDD33_ LDOIN							
V11	67	VSS			VSS							
U11	66	VSS_KEL0			VSS_KEL0							
W14	—	LVDS0P			LVDS0P							
Y14	—	LVDS0N			LVDS0N							
K4	3	JTCLK/ SWCLK	JTCLK/ SWCLK	PTA8	JTCLK/ SWCLK			DCU0_R0				
K2	4	JTDI	JTDI	PTA9	JTDI	RMII_CLKOUT	RMII_CLKIN/ MII0_TXCLK	DCU0_R1		WDOG_b		
K1	5	JTDO	JTDO/ TRACESWO	PTA10	JTDO	EXT_AUDIO_ MCLK		DCU0_G0		ENET_TS_ CLKIN		
L1	6	JTMS/ SWDIO	JTMS/ SWDIO	PTA11	JTMS/ SWDIO			DCU0_G1				
L3	7	PTA12		PTA12	TRACECK	EXT_AUDIO_ MCLK				VIU_DATA13	I2C0_SCL	
Y5	43	PTA16		PTA16	TRACED0	USB0_VBUS_ EN	ADC1_SE0	LCD29	SAI2_TX_ BCLK	VIU_DATA14	I2C0_SDA	
Y6	44	PTA17		PTA17	TRACED1	USB0_VBUS_ OC	ADC1_SE1	LCD30	USB0_SOF_ PULSE	VIU_DATA15	I2C1_SCL	
V6	46	PTA18		PTA18	TRACED2	ADC0_SE0	FTM1_QD_ PHA	LCD31	SAI2_TX_ DATA	VIU_DATA16	I2C1_SDA	
U6	47	PTA19		PTA19	TRACED3	ADC0_SE1	FTM1_QD_ PHB	LCD32	SAI2_TX_ SYNC	VIU_DATA17	QSPI1_A_SCK	
B18	143	PTA20		PTA20	TRACED4			LCD33		SCI3_TX	DCU1_ HSYNC/ DCU1_TCON1	
D18	145	PTA21		PTA21/ MII0_RXCLK	TRACED5				SAI2_RX_ BCLK	SCI3_RX	DCU1_ VSYNC/ DCU1_TCON2	
E17	147	PTA22		PTA22	TRACED6				SAI2_RX_ DATA	I2C2_SCL	DCU1_TAG/ DCU1_TCON0	

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
C17	148	PTA23		PTA23	TRACED7				SAI2_RX_ SYNC	I2C2_SDA	DCU1_DE/ DCU1_TCON3	
R16	—	PTA24		PTA24	TRACED8	USB1_VBUS_ EN			SDHC1_CLK	DCU1_TCON4		
R17	—	PTA25		PTA25	TRACED9	USB1_VBUS_ OC			SDHC1_CMD	DCU1_TCON5		
R19	—	PTA26		PTA26	TRACED10	SAI3_TX_ BCLK			SDHC1_DAT0	DCU1_TCON6		
R20	—	PTA27		PTA27	TRACED11	SAI3_RX_ BCLK			SDHC1_DAT1	DCU1_TCON7		
P20	—	PTA28		PTA28	TRACED12	SAI3_RX_ DATA	ENET1_1588_ TMR0	SCI4_TX	SDHC1_DAT2	DCU1_TCON8		
P18	—	PTA29		PTA29	TRACED13	SAI3_TX_ DATA	ENET1_1588_ TMR1	SCI4_RX	SDHC1_DAT3	DCU1_TCON9		
P17	—	PTA30		PTA30	TRACED14	SAI3_RX_ SYNC	ENET1_1588_ TMR2	SCI4_RTS	I2C3_SCL		SCI3_TX	
P16	—	PTA31		PTA31	TRACED15	SAI3_TX_ SYNC	ENET1_1588_ TMR3	SCI4_CTS	I2C3_SDA		SCI3_RX	
T6	49	PTB0		PTB0	FTM0_CH0	ADC0_SE2	TRACECTL	LCD34	SAI2_RX_ BCLK	VIU_DATA18	QSPI1_A_CS0	
T7	50	PTB1	RCON30	PTB1	FTM0_CH1	ADC0_SE3	RCON30	LCD35	SAI2_RX_ DATA	VIU_DATA19	QSPI1_A_ DATA3	
V7	51	PTB2	RCON31	PTB2	FTM0_CH2	ADC1_SE2	RCON31	LCD36	SAI2_RX_ SYNC	VIU_DATA20	QSPI1_A_ DATA2	
W7	53	PTB3		PTB3	FTM0_CH3	ADC1_SE3	EXTRIG	LCD37		VIU_DATA21	QSPI1_A_ DATA1	
Y7	54	PTB4		PTB4	FTM0_CH4	SCI1_TX	ADC0_SE4	LCD38	VIU_FID	VIU_DATA22	QSPI1_A_ DATA0	
Y8	55	PTB5		PTB5	FTM0_CH5	SCI1_RX	ADC1_SE4	LCD39	VIU_DE	VIU_DATA23	QSPI1_A_ DQS	
W8	56	PTB6		PTB6	FTM0_CH6	SCI1_RTS	QSPI0_A_CS1	LCD40	FB_CLKOUT	VIU_HSYNC	SCI2_TX	
D13	166	PTB7		PTB7	FTM0_CH7	SCI1_CTS	QSPI0_B_CS1	LCD41		VIU_VSYNC	SCI2_RX	
J16	121	PTB8		PTB8	FTM1CH0		FTM1_QD_ PHA		VIU_DE		DCU1_R6	
J19	123	PTB9		PTB9	FTM1CH1		FTM1_QD_ PHB				DCU1_R7	
B15	159	PTB10		PTB10	SCI0_TX			DCU0_TCON4	VIU_DE	CKO1	ENET_TS_ CLKIN	
D14	164	PTB11		PTB11	SCI0_RX			DCU0_TCON5	SNVS_ ALARM_OUT_ B	CKO2	ENET0_1588_ TMR0	
E13	165	PTB12	NMI	PTB12	SCI0_RTS		SPI0_PCS5	DCU0_TCON6	FB_AD1	NMI	ENET0_1588_ TMR1	
D15	156	PTB13		PTB13	SCI0_CTS		SPI0_PCS4	DCU0_TCON7	FB_AD0	TRACECTL		
B14	162	PTB14		PTB14	CAN0_RX	I2C0_SCL		DCU0_TCON8			DCU1_PCLK	
A14	161	PTB15		PTB15	CAN0_TX	I2C0_SDA		DCU0_TCON9			VIU_PIX_CLK	

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
C14	163	PTB16		PTB16	CAN1_RX	I2C1_SCL		DCU0_ TCON10				
A15	160	PTB17		PTB17	CAN1_TX	I2C1_SDA		DCU0_ TCON11				
B12	171	PTB18		PTB18	SPI0_PCS1	EXT_AUDIO_ MCLK		CKO1		VIU_DATA9	CCM_OBS0	
C13	167	PTB19		PTB19	SPI0_PCS0					VIU_DATA10	CCM_OBS1	
A13	169	PTB20		PTB20	SPI0_SIN			LCD42		VIU_DATA11	CCM_OBS2	
E12	173	PTB21		PTB21	SPI0_SOUT			LCD43		VIU_DATA12	DCU1_PCLK	
D12	172	PTB22		PTB22	SPI0_SCK				VIU_FID			
V10	61	USB0_GND			USB0_GND							
T10	63	USB0_DP			USB0_DP							
T9	62	USB0_DM			USB0_DM							
W11	60	USB0_VBUS			USB0_VBUS							
Y10	59	USB_DCAP			USB_DCAP							
Y11	64	USB0_VBUS_ DETECT			USB0_VBUS_ DETECT							
Y9	—	USB1_GND			USB1_GND							
W9	—	USB1_DP			USB1_DP							
V9	—	USB1_DM			USB1_DM							
W10	—	USB1_VBUS			USB1_VBUS							
U9	—	USB1_VBUS_ DETECT			USB1_VBUS_ DETECT							
L4	8	PTC0		PTC0	RMII0_MDC/ MII0_MDC	FTM1CH0	SPI0_PCS3	ESAI_SCKT	SDHC0_CLK	VIU_DATA0	RCON18	
L5	9	PTC1		PTC1	RMII0_MDIO/ MII0_MDC	FTM1CH1	SPI0_PCS2	ESAI_FST	SDHC0_CMD	VIU_DATA1	RCON19	
M5	11	PTC2		PTC2	RMII0_CRS_ DV	SCI1_TX		ESAI_SDO0	SDHC0_DAT0	VIU_DATA2	RCON20	
M3	12	PTC3		PTC3	RMII0_RXD1/ MII0_RXD[1]	SCI1_RX		ESAI_SDO1	SDHC0_DAT1	VIU_DATA3	DCU0_R0	
L2	14	PTC4		PTC4	RMII0_RXD0/ MII0_RXD[0]	SCI1_RTS	SPI1_PCS1	ESAI_SDO2/ ESAI_SDI3	SDHC0_DAT2	VIU_DATA4	DCU0_R1	
M1	15	PTC5		PTC5	RMII0_RXER/ MII0_RXER	SCI1_CTS	SPI1_PCS0	ESAI_SDO3/ ESAI_SDI2	SDHC0_DAT3	VIU_DATA5	DCU0_G0	
N1	16	PTC6		PTC6	RMII0_TXD1/ MII0_TXD[1]		SPI1_SIN	ESAI_SDO5/ ESAI_SDI0	SDHC0_WP	VIU_DATA6	DCU0_G1	
N2	17	PTC7		PTC7	RMII0_TXD0/ MII0_TXD[0]		SPI1_SOUT	ESAI_SDO4/ ESAI_SDI1		VIU_DATA7	DCU0_B0	
N4	18	PTC8		PTC8	RMII0_TXEN/ MII0_TXEN		SPI1_SCK			VIU_DATA8	DCU0_B1	
T15	77	PTC9		PTC9	RMII1_MDC		ESAI_SCKT					
U15	78	PTC10		PTC10	RMII1_MDIO		ESAI_FST					
P4	20	PTC11		PTC11	RMII1_CRS_ DV		ESAI_SDO0					

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
P3	21	PTC12		PTC12	RMII1_RXD1		ESAI_SDO1		SAI2_TX_BCLK			
P1	23	PTC13		PTC13	RMII1_RXD0		ESAI_SDO2/ESAI_SDI3		SAI2_RX_BCLK			
R1	26	PTC14		PTC14	RMII1_RXER		ESAI_SDO3/ESAI_SDI2	SCI5_TX	SAI2_RX_DATA	ADC0_SE6		
P2	27	PTC15		PTC15	RMII1_TXD1		ESAI_SDI0	SCI5_RX	SAI2_TX_DATA	ADC0_SE7		
R3	29	PTC16		PTC16	RMII1_TXD0		ESAI_SDO4/ESAI_SDI1	SCI5_RTS	SAI2_RX_SYNC	ADC1_SE6		
R4	28	PTC17		PTC17	RMII1_TXEN		ADC1_SE7	SCI5_CTS	SAI2_TX_SYNC	USB1_SOF_PULSE		
B10	—	DDR_A[15]			DDR_A15							
D9	—	DDR_A[14]			DDR_A14							
A10	—	DDR_A[13]			DDR_A13							
C10	—	DDR_A[12]			DDR_A12							
D10	—	DDR_A[11]			DDR_A11							
D7	—	DDR_A[10]			DDR_A10							
B9	—	DDR_A[9]			DDR_A9							
A11	—	DDR_A[8]			DDR_A8							
A7	—	DDR_A[7]			DDR_A7							
A9	—	DDR_A[6]			DDR_A6							
B6	—	DDR_A[5]			DDR_A5							
A6	—	DDR_A[4]			DDR_A4							
B7	—	DDR_A[3]			DDR_A3							
A8	—	DDR_A[2]			DDR_A2							
C11	—	DDR_A[1]			DDR_A1							
C7	—	DDR_A[0]			DDR_A0							
D8	—	DDR_BA[2]			DDR_BA2							
C9	—	DDR_BA[1]			DDR_BA1							
C8	—	DDR_BA[0]			DDR_BA0							
B4	—	DDR_CAS_b			DDR_CAS_b							
A5	—	DDR_CKE[0]			DDR_CKE0							
A2	—	DDR_CLK[0]			DDR_CLK0							
B2	—	DDR_CLK_b[0]			DDR_CLK_b0							
C5	—	DDR_CS_b[0]			DDR_CS_b0							
D2	—	DDR_D[15]			DDR_D15							
H2	—	DDR_D[14]			DDR_D14							
C1	—	DDR_D[13]			DDR_D13							
G1	—	DDR_D[12]			DDR_D12							
E2	—	DDR_D[11]			DDR_D11							

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
H1	—	DDR_D[10]			DDR_D10							
D1	—	DDR_D[9]			DDR_D9							
J1	—	DDR_D[8]			DDR_D8							
G3	—	DDR_D[7]			DDR_D7							
C3	—	DDR_D[6]			DDR_D6							
J3	—	DDR_D[5]			DDR_D5							
F3	—	DDR_D[4]			DDR_D4							
G4	—	DDR_D[3]			DDR_D3							
D4	—	DDR_D[2]			DDR_D2							
H3	—	DDR_D[1]			DDR_D1							
F4	—	DDR_D[0]			DDR_D0							
G2	—	DDR_DQM[1]			DDR_DQM1							
J4	—	DDR_DQM[0]			DDR_DQM0							
E1	—	DDR_DQS[1]			DDR_DQS1							
D3	—	DDR_DQS[0]			DDR_DQS0							
F1	—	DDR_DQS_ b[1]			DDR_DQS_b1							
E3	—	DDR_DQS_ b[0]			DDR_DQS_b0							
A4	—	DDR_RAS_b			DDR_RAS_b							
C6	—	DDR_WE_b			DDR_WE_b							
C4	—	DDR_ODT[0]			DDR_ODT0							
B1	—	DDR_ODT[1]			DDR_ODT1							
G5	—	DDR_VREF			DDR_VREF							
A3	—	DDR_ZQ			DDR_ZQ							
D6	—	DDR_RESET			DDR_RESET							
J20	—	PTD31		PTD31	FB_AD31	NF_IO15		FTM3_CH0	SPI2_PCS1			
H20	—	PTD30		PTD30	FB_AD30	NF_IO14		FTM3_CH1	SPI2_PCS0			
H18	—	PTD29		PTD29	FB_AD29	NF_IO13		FTM3_CH2	SPI2_SIN			
H17	—	PTD28		PTD28	FB_AD28	NF_IO12	I2C2_SCL	FTM3_CH3	SPI2_SOUT			
H16	—	PTD27		PTD27	FB_AD27	NF_IO11	I2C2_SDA	FTM3_CH4	SPI2_SCK			
G16	—	PTD26		PTD26	FB_AD26	NF_IO10		FTM3_CH5	SDHC1_WP			
G18	—	PTD25		PTD25	FB_AD25	NF_IO9		FTM3_CH6				
G19	—	PTD24		PTD24	FB_AD24	NF_IO8		FTM3_CH7				
G20	124	PTD23		PTD23/ MII0_ RXDATA[3]	FB_AD23	NF_IO7	FTM2CH0	ENET0_1588_ TMR0	SDHC0_DAT4	SCI2_TX	DCU1_R3	
F20	126	PTD22		PTD22/ MII0_ RXDATA[2]	FB_AD22	NF_IO6	FTM2CH1	ENET0_1588_ TMR1	SDHC0_DAT5	SCI2_RX	DCU1_R4	
F19	128	PTD21		PTD21/ MII0_CRS	FB_AD21	NF_IO5		ENET0_1588_ TMR2	SDHC0_DAT6	SCI2_RTS	DCU1_R5	

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
F17	129	PTD20		PTD20/ MII0_COL	FB_AD20	NF_IO4		ENET0_1588_ TMR3	SDHC0_DAT7	SCI2_CTS	DCU1_R0	
F16	130	PTD19		PTD19	FB_AD19	NF_IO3	ESAI_SCKR	I2C0_SCL	FTM2_QD_ PHA	MII0_ TXDATA[3]	DCU1_R1	
E18	131	PTD18		PTD18	FB_AD18	NF_IO2	ESAI_FSR	I2C0_SDA	FTM2_QD_ PHB	MII0_ TXDATA[2]	DCU1_G0	
E20	132	PTD17		PTD17	FB_AD17	NF_IO1	ESAI_HCKR	I2C1_SCL		MII0_TXERR	DCU1_G1	
D20	133	PTD16		PTD16	FB_AD16	NF_IO0	ESAI_HCKT	I2C1_SDA			DCU1_G2	
Y17	86	PTD0		PTD0	QSPI0_A_SCK	SCI2_TX		FB_AD15	SPDIF_ EXTCLK			
Y18	87	PTD1		PTD1	QSPI0_A_CS0	SCI2_RX		FB_AD14	SPDIF_IN1			
V18	88	PTD2		PTD2	QSPI0_A_ DATA3	SCI2_RTS	SPI1_PCS3	FB_AD13	SPDIF_OUT1			
Y19	89	PTD3		PTD3	QSPI0_A_ DATA2	SCI2_CTS	SPI1_PCS2	FB_AD12	SPDIF_ PLOCK			
W19	90	PTD4		PTD4	QSPI0_A_ DATA1		SPI1_PCS1	FB_AD11	SPDIF_ SRCLK			
W20	91	PTD5		PTD5	QSPI0_A_ DATA0		SPI1_PCS0	FB_AD10				
V20	92	PTD6		PTD6	QSPI0_A_ DQS		SPI1_SIN	FB_AD9				
V19	93	PTD7		PTD7	QSPI0_B_SCK		SPI1_SOUT	FB_AD8				
U17	94	PTD8		PTD8	QSPI0_B_CS0	FB_CLKOUT	SPI1_SCK	FB_AD7				
U18	97	PTD9		PTD9	QSPI0_B_ DATA3	SPI3_PCS1		FB_AD6		SAI1_TX_ SYNC	DCU1_B0	
U20	98	PTD10		PTD10	QSPI0_B_ DATA2	SPI3_PCS0		FB_AD5			DCU1_B1	
T20	99	PTD11		PTD11	QSPI0_B_ DATA1	SPI3_SIN		FB_AD4				
T19	100	PTD12		PTD12	QSPI0_B_ DATA0	SPI3_SOUT		FB_AD3				
T18	101	PTD13		PTD13	QSPI0_B_ DQS	SPI3_SCK		FB_AD2				
A19	141	PTB23		PTB23	SAI0_TX_ BCLK	SCI1_TX		FB_MUXED_ ALE	FB_TS_b	SCI3_RTS	DCU1_G3	
A18	142	PTB24		PTB24	SAI0_RX_ BCLK	SCI1_RX		FB_MUXED_ TSIZ0	NF_WE_b	SCI3_CTS	DCU1_G4	
B17	149	PTB25		PTB25	SAI0_RX_ DATA	SCI1_RTS		FB_CS1_b	NF_CE0_b		DCU1_G5	
A17	150	PTB26	RCON21	PTB26	SAI0_TX_ DATA	SCI1_CTS	RCON21	FB_CS0_b	NF_CE1_b		DCU1_G6	
U8	57	PTB27	RCON22	PTB27	SAI0_RX_ SYNC		RCON22	FB_OE_b	FB_MUXED_ TBST_b	NF_RE_b	DCU1_G7	
A16	151	PTB28	RCON23	PTB28	SAI0_TX_ SYNC		RCON23	FB_RW_b			DCU1_B6	

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
D16	153	PTC26	RCON24	PTC26	SAI1_TX_ BCLK	SPI0_PCS5	RCON24	FB_TA_b	NF_RB_b		DCU1_B7	
E16	154	PTC27	RCON25	PTC27	SAI1_RX_ BCLK	SPI0_PCS4	RCON25	FB_BE3_b	FB_CS3_b	NF_ALE	DCU1_B2	
E15	155	PTC28	RCON26	PTC28	SAI1_RX_ DATA	SPI0_PCS3	RCON26	FB_BE2_b	FB_CS2_b	NF_CLE	DCU1_B3	
C16	152	PTC29	RCON27	PTC29	SAI1_TX_ DATA	SPI0_PCS2	RCON27	FB_BE1_b	FB_MUXED_ TSIZ1		DCU1_B4	
T8	58	PTC30	RCON28	PTC30	SAI1_RX_ SYNC	SPI1_PCS2	RCON28	FB_MUXED_ BE0_b	FB_TSIZ0	ADC0_SE5	DCU1_B5	
W5	42	PTC31	RCON29	PTC31	SAI1_TX_ SYNC		RCON29			ADC1_SE5	DCU1_B6	
N16	103	PTE0	BOOTMOD1	PTE0	DCU0_ HSYNC/ DCU0_TCON1	BOOTMOD1		LCD0				
N18	104	PTE1	BOOTMOD0	PTE1	DCU0_ VSYNC/ DCU0_TCON2	BOOTMOD0		LCD1				
N19	105	PTE2		PTE2	DCU0_PCLK			LCD2				
Y15	80	PTE3		PTE3	DCU0_TAG/ DCU0_TCON0			LCD3				
N20	106	PTE4		PTE4	DCU0_DE/ DCU0_TCON3			LCD4				
T16	—	PTE5		PTE5	DCU0_R0			LCD5				
W16	—	PTE6		PTE6	DCU0_R1			LCD6				
M20	109	PTE7	RCON0	PTE7	DCU0_R2		RCON0	LCD7				
M19	110	PTE8	RCON1	PTE8	DCU0_R3		RCON1	LCD8				
M17	111	PTE9	RCON2	PTE9	DCU0_R4		RCON2	LCD9				
M16	112	PTE10	RCON3	PTE10	DCU0_R5		RCON3	LCD10				
L16	113	PTE11	RCON4	PTE11	DCU0_R6		RCON4	LCD11				
L17	114	PTE12	RCON5	PTE12	DCU0_R7	SPI1_PCS3	RCON5	LCD12			LPT_ALT0	
Y16	—	PTE13		PTE13	DCU0_G0			LCD13				
W15	—	PTE14		PTE14	DCU0_G1			LCD14				
L18	115	PTE15	RCON6	PTE15	DCU0_G2		RCON6	LCD15				
L20	116	PTE16	RCON7	PTE16	DCU0_G3		RCON7	LCD16				
K20	117	PTE17	RCON8	PTE17	DCU0_G4		RCON8	LCD17				
K19	118	PTE18	RCON9	PTE18	DCU0_G5		RCON9	LCD18				
K18	119	PTE19	RCON10	PTE19	DCU0_G6		RCON10	LCD19	I2C0_SCL			
A12	170	PTE20	RCON11	PTE20	DCU0_G7		RCON11	LCD20	I2C0_SDA		EWM_in	
V16	81	PTE21		PTE21	DCU0_B0			LCD21				
W17	84	PTE22		PTE22	DCU0_B1			LCD22				
J17	122	PTE23	RCON12	PTE23	DCU0_B2		RCON12	LCD23				
D19	134	PTE24	RCON13	PTE24	DCU0_B3		RCON13	LCD24				
C19	135	PTE25	RCON14	PTE25	DCU0_B4		RCON14	LCD25				

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
C20	137	PTE26	RCON15	PTE26	DCU0_B5		RCON15	LCD26				
B20	138	PTE27	RCON16	PTE27	DCU0_B6		RCON16	LCD27	I2C1_SCL			
K16	120	PTE28	RCON17	PTE28	DCU0_B7		RCON17	LCD28	I2C1_SDA		EWM_out	
V15	79	PTA7		PTA7	VIU_PIX_CLK							
T14	76	EXT_ TAMPER0			EXT_ TAMPER0							
U14	74	EXT_ TAMPER1			EXT_ TAMPER1							
T13	—	EXT_ TAMPER2/ EXT_WM0_ TAMPER_IN			EXT_ TAMPER2/ EXT_WM0_ TAMPER_IN							
U13	—	EXT_ TAMPER3/ EXT_WM0_ TAMPER_ OUT			EXT_ TAMPER3/ EXT_WM0_ TAMPER_ OUT							
U12	—	EXT_ TAMPER4/ EXT_WM1_ TAMPER_IN			EXT_ TAMPER4/ EXT_WM1_ TAMPER_IN							
U10	—	EXT_ TAMPER5/ EXT_WM1_ TAMPER_ OUT			EXT_ TAMPER5/ EXT_WM1_ TAMPER_ OUT							
G7	2	VDD			VDD							
J7	—	VDD			VDD							
L7	22	VDD			VDD							
H8	48	VDD			VDD							
K8	85	VDD			VDD							
M8	102	VDD			VDD							
P8	125	VDD			VDD							
G9	136	VDD			VDD							
N9	174	VDD			VDD							
H10	—	VDD			VDD							
P10	—	VDD			VDD							
G11	—	VDD			VDD							
N11	—	VDD			VDD							
H12	—	VDD			VDD							
P12	—	VDD			VDD							
G13	—	VDD			VDD							
J13	—	VDD			VDD							
L13	—	VDD			VDD							
N13	—	VDD			VDD							

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
H14	—	VDD			VDD							
K14	—	VDD			VDD							
M14	—	VDD			VDD							
P14	—	VDD			VDD							
A1	1	VSS			VSS							
A20	13	VSS			VSS							
B3	24	VSS			VSS							
B5	32	VSS			VSS							
B8	—	VSS			VSS							
B11	—	VSS			VSS							
B13	—	VSS			VSS							
B16	—	VSS			VSS							
B19	—	VSS			VSS							
C2	—	VSS			VSS							
D17	—	VSS			VSS							
E5	—	VSS			VSS							
E8	—	VSS			VSS							
E11	—	VSS			VSS							
E14	—	VSS			VSS							
E19	—	VSS			VSS							
F2	—	VSS			VSS							
G17	—	VSS			VSS							
H4	—	VSS			VSS							
J2	—	VSS			VSS							
J18	—	VSS			VSS							
M2	—	VSS			VSS							
M4	—	VSS			VSS							
M18	—	VSS			VSS							
R2	—	VSS			VSS							
R18	—	VSS			VSS							
U7	—	VSS			VSS							
U19	—	VSS			VSS							
V13	—	VSS			VSS							
W6	—	VSS			VSS							
V17	—	VSS			VSS							
Y1	—	VSS			VSS							
Y20	—	VSS			VSS							
H19	—	VSS			VSS							
L19	—	VSS			VSS							
P19	—	VSS			VSS							

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
J5	—	SDRAMC_ VDD2P5			SDRAMC_ VDD2P5							
E6	—	SDRAMC_ VDD2P5			SDRAMC_ VDD2P5							
E10	—	SDRAMC_ VDD2P5			SDRAMC_ VDD2P5							
E4	—	SDRAMC_ VDD1P5			SDRAMC_ VDD1P5							
D5	—	SDRAMC_ VDD1P5			SDRAMC_ VDD1P5							
F5	—	SDRAMC_ VDD1P5			SDRAMC_ VDD1P5							
H5	—	SDRAMC_ VDD1P5			SDRAMC_ VDD1P5							
K5	—	SDRAMC_ VDD1P5			SDRAMC_ VDD1P5							
E7	—	SDRAMC_ VDD1P5			SDRAMC_ VDD1P5							
E9	—	SDRAMC_ VDD1P5			SDRAMC_ VDD1P5							
D11	—	SDRAMC_ VDD1P5			SDRAMC_ VDD1P5							
K3	10	VDD33			VDD33							
N3	25	VDD33			VDD33							
V8	52	VDD33			VDD33							
C12	83	VDD33			VDD33							
C15	—	VDD33			VDD33							
U16	95	VDD33			VDD33							
K17	108	VDD33			VDD33							
N17	127	VDD33			VDD33							
T17	140	VDD33			VDD33							
C18	146	VDD33			VDD33							
F18	158	VDD33			VDD33							
W18	168	VDD33			VDD33							
H7	—	VSS			VSS							
K7	45	VSS			VSS							
M7	82	VSS			VSS							
P7	—	VSS			VSS							
G8	96	VSS			VSS							
J8	107	VSS			VSS							
L8	—	VSS			VSS							
N8	139	VSS			VSS							
H9	144	VSS			VSS							

364 MAP BGA	176 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
J9	157	VSS			VSS							
K9	175	VSS			VSS							
L9	176	VSS			VSS							
M9	—	VSS			VSS							
P9	—	VSS			VSS							
G10	—	VSS			VSS							
J10	—	VSS			VSS							
K10	—	VSS			VSS							
L10	—	VSS			VSS							
M10	—	VSS			VSS							
N10	—	VSS			VSS							
H11	—	VSS			VSS							
J11	—	VSS			VSS							
K11	—	VSS			VSS							
L11	—	VSS			VSS							
M11	—	VSS			VSS							
P11	—	VSS			VSS							
G12	—	VSS			VSS							
J12	—	VSS			VSS							
K12	—	VSS			VSS							
L12	—	VSS			VSS							
M12	—	VSS			VSS							
N12	—	VSS			VSS							
H13	—	VSS			VSS							
K13	—	VSS			VSS							
M13	—	VSS			VSS							
P13	—	VSS			VSS							
G14	—	VSS			VSS							
J14	—	VSS			VSS							
L14	—	VSS			VSS							
N14	—	VSS			VSS							
N7	—	FA_VDD			FA_VDD							
V14	75	VBAT			VBAT							

4.2 Pinout diagrams

NOTE

If tamper detection is not required, the tamper pins must be tied to ground.

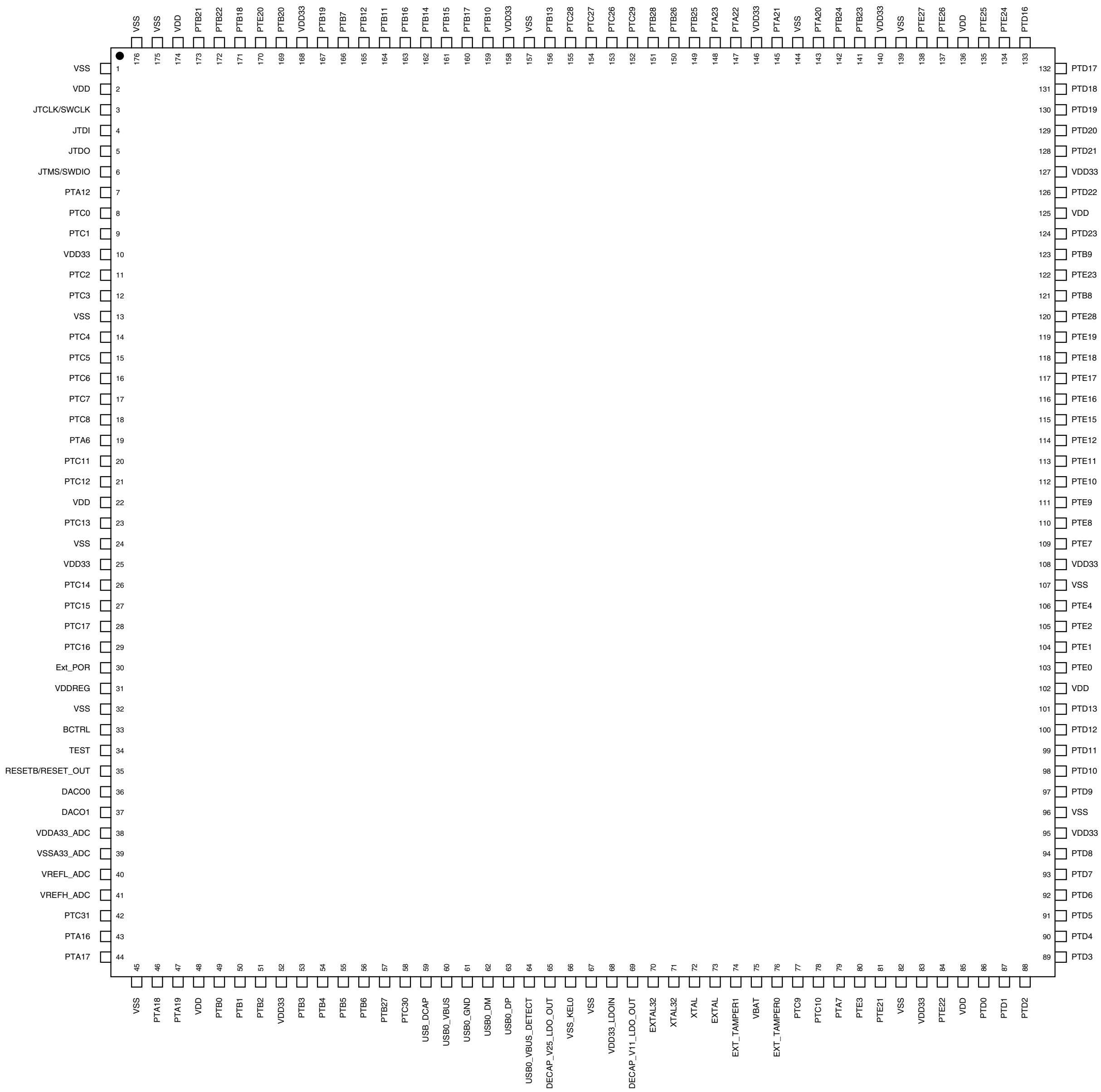


Figure 4-1. 176 LQFP Pinout Diagram

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	VSS	DDR_CLK[0]	DDR_ZQ	DDR_RAS_b	DDR_CKE[0]	DDR_A[4]	DDR_A[7]	DDR_A[2]	DDR_A[6]	DDR_A[13]	DDR_A[8]	PTE20	PTB20	PTB15	PTB17	PTB28	PTB26	PTB24	PTB23	VSS
B	DDR_ODT[1]	DDR_CLK_b[0]	VSS	DDR_CAS_b	VSS	DDR_A[5]	DDR_A[3]	VSS	DDR_A[9]	DDR_A[15]	VSS	PTB18	VSS	PTB14	PTB10	VSS	PTB25	PTA20	VSS	PTE27
C	DDR_D[13]	VSS	DDR_D[6]	DDR_ODT[0]	DDR_CS_b[0]	DDR_WE_b	DDR_A[0]	DDR_BA[0]	DDR_BA[1]	DDR_A[12]	DDR_A[1]	VDD33	PTB19	PTB16	VDD33	PTC29	PTA23	VDD33	PTE25	PTE26
D	DDR_D[9]	DDR_D[15]	DDR_DQS[0]	DDR_D[2]	SDRAMC_VDD1P5	DDR_RESET	DDR_A[10]	DDR_BA[2]	DDR_A[14]	DDR_A[11]	SDRAMC_VDD1P5	PTB22	PTB7	PTB11	PTB13	PTC26	VSS	PTA21	PTE24	PTD16
E	DDR_DQS[1]	DDR_D[11]	DDR_DQS_b[0]	SDRAMC_VDD1P5	VSS	SDRAMC_VDD2P5	SDRAMC_VDD1P5	VSS	SDRAMC_VDD1P5	SDRAMC_VDD2P5	VSS	PTB21	PTB12	VSS	PTC28	PTC27	PTA22	PTD18	VSS	PTD17
F	DDR_DQS_b[1]	VSS	DDR_D[4]	DDR_D[0]	SDRAMC_VDD1P5											PTD19	PTD20	VDD33	PTD21	PTD22
G	DDR_D[12]	DDR_DQM[1]	DDR_D[7]	DDR_D[3]	DDR_VREF		VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS		PTD26	VSS	PTD25	PTD24	PTD23
H	DDR_D[10]	DDR_D[14]	DDR_D[1]	VSS	SDRAMC_VDD1P5		VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD		PTD27	PTD28	PTD29	VSS	PTD30
J	DDR_D[8]	VSS	DDR_D[5]	DDR_DQM[0]	SDRAMC_VDD2P5		VDD	VSS	VSS	VSS	VSS	VSS	VDD	VSS		PTB8	PTE23	VSS	PTB9	PTD31
K	JTDO	JTDI	VDD33	JTCLK/SWCLK	SDRAMC_VDD1P5		VSS	VDD	VSS	VSS	VSS	VSS	VSS	VDD		PTE28	VDD33	PTE19	PTE18	PTE17
L	JTMS/SWDIO	PTC4	PTA12	PTC0	PTC1		VDD	VSS	VSS	VSS	VSS	VSS	VDD	VSS		PTE11	PTE12	PTE15	VSS	PTE16
M	PTC5	VSS	PTC3	VSS	PTC2		VSS	VDD	VSS	VSS	VSS	VSS	VSS	VDD		PTE10	PTE9	VSS	PTE8	PTE7
N	PTC6	PTC7	VDD33	PTC8	PTA6		FA_VDD	VSS	VDD	VSS	VDD	VSS	VDD	VSS		PTE0	VDD33	PTE1	PTE2	PTE4
P	PTC13	PTC15	PTC12	PTC11	VDDREG		VSS	VDD	VSS	VDD	VSS	VDD	VSS	VDD		PTA31	PTA30	PTA29	VSS	PTA28
R	PTC14	VSS	PTC16	PTC17	VSS12_AFE											PTA24	PTA25	VSS	PTA26	PTA27
T	TEST2	BCTRL	TEST	RESETB/RESET_OUT	VDD12_AFE	PTB0	PTB1	PTC30	USB0_DM	USB0_DP	DECAP_V25_LDO_OUT	VDD33_LDOIN	EXT_TAMPER2/EXT_WM0_TAMPER_IN	EXT_TAMPER0	PTC9	PTE5	VDD33	PTD13	PTD12	PTD11
U	DACO0	DACO1	VREFL_ADC	VADCSE1	VADC_AFE_BANDGAP	PTA19	VSS	PTB27	USB1_VBUS_DETECT	EXT_TAMPER5/EXT_WM1_TAMPER_OUT	VSS_KEL0	EXT_TAMPER4/EXT_WM1_TAMPER_IN	EXT_TAMPER3/EXT_WM0_TAMPER_OUT	EXT_TAMPER1	PTC10	VDD33	PTD8	PTD9	VSS	PTD10
V	VDDA33_ADC	VSSA33_ADC	VDDA33_AFE	VSSA33_AFE	VADCSE3	PTA18	PTB2	VDD33	USB1_DM	USB0_GND	VSS	DECAP_V11_LDO_OUT	VSS	VBAT	PTA7	PTE21	VSS	PTD2	PTD7	PTD6
W	VREFH_ADC	ADC0SE9	ADC1SE8	VADCSE2	PTC31	VSS	PTB3	PTB6	USB1_DP	USB1_VBUS	USB0_VBUS	XTAL32	XTAL	LVDS0P	PTE14	PTE6	PTE22	VDD33	PTD4	PTD5
Y	VSS	ADC0SE8	ADC1SE9	VADCSE0	PTA16	PTA17	PTB4	PTB5	USB1_GND	USB_DCAP	USB0_VBUS_DETECT	EXTAL32	EXTAL	LVDS0N	PTE3	PTE13	PTD0	PTD1	PTD3	VSS

Figure 4-2. 364-pin BGA package ballmap

Chapter 5
Input/Output Multiplexer Controller (IOMUXC)

5.1 Overview

The IOMUX Controller (IOMUXC), together with the IOMUX, enables the device to share one pad to several functional blocks. The sharing is done by multiplexing the pad input/output signals.

Every module requires a specific pad setting (such as pull up, keeper, and so on). The pad settings parameters are controlled by the IOMUXC.

The IOMUX consists only of combinatorial logic combined from several basic iomux cells. Each basic iomux cell handles only one pad signal's muxing.

NOTE

IOMUX registers control the PAD settings. To see the PAD mapping on this device with the PORT pins, refer to GPIO Mapping

NOTE

For TCON[x] signals, like tcon0 refer to DCU0_TCONx signal and tcon1 refer to DCU1_TCONx signal in the Pinouts table.

5.2 Memory map and register definition

IOMUXC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_8000	Software MUX Pad Control Register 0 (IOMUXC_PTA6)	32	R/W	0000_0060h	5.2.1/152
4004_8004	Software MUX Pad Control Register 1 (IOMUXC_PTA8)	32	R/W	0010_006Dh	5.2.2/154
4004_8008	Software MUX Pad Control Register 2 (IOMUXC_PTA9)	32	R/W	0010_006Dh	5.2.3/155

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_800C	Software MUX Pad Control Register 3 (IOMUXC_PTA10)	32	R/W	0010_3060h	5.2.4/157
4004_8010	Software MUX Pad Control Register 4 (IOMUXC_PTA11)	32	R/W	0010_006Dh	5.2.5/159
4004_8014	Software MUX Pad Control Register 5 (IOMUXC_PTA12)	32	R/W	0000_0060h	5.2.6/161
4004_8018	Software MUX Pad Control Register 6 (IOMUXC_PTA16)	32	R/W	0000_0060h	5.2.7/162
4004_801C	Software MUX Pad Control Register 7 (IOMUXC_PTA17)	32	R/W	0000_0060h	5.2.8/164
4004_8020	Software MUX Pad Control Register 8 (IOMUXC_PTA18)	32	R/W	0000_0060h	5.2.9/166
4004_8024	Software MUX Pad Control Register 9 (IOMUXC_PTA19)	32	R/W	0000_0060h	5.2.10/168
4004_8028	Software MUX Pad Control Register 10 (IOMUXC_PTA20)	32	R/W	0000_0060h	5.2.11/169
4004_802C	Software MUX Pad Control Register 11 (IOMUXC_PTA21)	32	R/W	0000_0060h	5.2.12/171
4004_8030	Software MUX Pad Control Register 12 (IOMUXC_PTA22)	32	R/W	0000_0060h	5.2.13/173
4004_8034	Software MUX Pad Control Register 13 (IOMUXC_PTA23)	32	R/W	0000_0060h	5.2.14/175
4004_8038	Software MUX Pad Control Register 14 (IOMUXC_PTA24)	32	R/W	0000_0060h	5.2.15/176
4004_803C	Software MUX Pad Control Register 15 (IOMUXC_PTA25)	32	R/W	0000_0060h	5.2.16/178
4004_8040	Software MUX Pad Control Register 16 (IOMUXC_PTA26)	32	R/W	0000_0060h	5.2.17/180
4004_8044	Software MUX Pad Control Register 17 (IOMUXC_PTA27)	32	R/W	0000_0060h	5.2.18/181
4004_8048	Software MUX Pad Control Register 18 (IOMUXC_PTA28)	32	R/W	0000_0060h	5.2.19/183
4004_804C	Software MUX Pad Control Register 19 (IOMUXC_PTA29)	32	R/W	0000_0060h	5.2.20/185
4004_8050	Software MUX Pad Control Register 20 (IOMUXC_PTA30)	32	R/W	0000_0060h	5.2.21/186
4004_8054	Software MUX Pad Control Register 21 (IOMUXC_PTA31)	32	R/W	0000_0060h	5.2.22/188
4004_8058	Software MUX Pad Control Register 22 (IOMUXC_PTB0)	32	R/W	0000_0060h	5.2.23/190
4004_805C	Software MUX Pad Control Register 23 (IOMUXC_PTB1)	32	R/W	0030_0060h	5.2.24/192
4004_8060	Software MUX Pad Control Register 24 (IOMUXC_PTB2)	32	R/W	0030_0060h	5.2.25/193
4004_8064	Software MUX Pad Control Register 25 (IOMUXC_PTB3)	32	R/W	0000_0060h	5.2.26/195
4004_8068	Software MUX Pad Control Register 26 (IOMUXC_PTB4)	32	R/W	0000_0060h	5.2.27/197
4004_806C	Software MUX Pad Control Register 27 (IOMUXC_PTB5)	32	R/W	0000_0060h	5.2.28/199
4004_8070	Software MUX Pad Control Register 28 (IOMUXC_PTB6)	32	R/W	0000_0060h	5.2.29/200
4004_8074	Software MUX Pad Control Register 29 (IOMUXC_PTB7)	32	R/W	0000_0060h	5.2.30/202
4004_8078	Software MUX Pad Control Register 30 (IOMUXC_PTB8)	32	R/W	0000_0060h	5.2.31/204
4004_807C	Software MUX Pad Control Register 31 (IOMUXC_PTB9)	32	R/W	0000_0060h	5.2.32/206
4004_8080	Software MUX Pad Control Register 32 (IOMUXC_PTB10)	32	R/W	0000_0060h	5.2.33/207
4004_8084	Software MUX Pad Control Register 33 (IOMUXC_PTB11)	32	R/W	0000_0060h	5.2.34/209
4004_8088	Software MUX Pad Control Register 34 (IOMUXC_PTB12)	32	R/W	0000_0060h	5.2.35/211
4004_808C	Software MUX Pad Control Register 35 (IOMUXC_PTB13)	32	R/W	0000_0060h	5.2.36/213
4004_8090	Software MUX Pad Control Register 36 (IOMUXC_PTB14)	32	R/W	0000_0060h	5.2.37/214
4004_8094	Software MUX Pad Control Register 37 (IOMUXC_PTB15)	32	R/W	0000_0060h	5.2.38/216
4004_8098	Software MUX Pad Control Register 38 (IOMUXC_PTB16)	32	R/W	0000_0060h	5.2.39/218
4004_809C	Software MUX Pad Control Register 39 (IOMUXC_PTB17)	32	R/W	0000_0060h	5.2.40/220
4004_80A0	Software MUX Pad Control Register 40 (IOMUXC_PTB18)	32	R/W	0000_0061h	5.2.41/221

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_80A4	Software MUX Pad Control Register 41 (IOMUXC_PTB19)	32	R/W	0000_0061h	5.2.42/223
4004_80A8	Software MUX Pad Control Register 42 (IOMUXC_PTB20)	32	R/W	0000_0060h	5.2.43/225
4004_80AC	Software MUX Pad Control Register 43 (IOMUXC_PTB21)	32	R/W	0000_0060h	5.2.44/226
4004_80B0	Software MUX Pad Control Register 44 (IOMUXC_PTB22)	32	R/W	0000_0061h	5.2.45/228
4004_80B4	Software MUX Pad Control Register 45 (IOMUXC_PTC0)	32	R/W	0070_0061h	5.2.46/230
4004_80B8	Software MUX Pad Control Register 46 (IOMUXC_PTC1)	32	R/W	0070_0061h	5.2.47/231
4004_80BC	Software MUX Pad Control Register 47 (IOMUXC_PTC2)	32	R/W	0070_0061h	5.2.48/233
4004_80C0	Software MUX Pad Control Register 48 (IOMUXC_PTC3)	32	R/W	0000_0060h	5.2.49/235
4004_80C4	Software MUX Pad Control Register 49 (IOMUXC_PTC4)	32	R/W	0000_0060h	5.2.50/237
4004_80C8	Software MUX Pad Control Register 50 (IOMUXC_PTC5)	32	R/W	0000_0060h	5.2.51/238
4004_80CC	Software MUX Pad Control Register 51 (IOMUXC_PTC6)	32	R/W	0000_0060h	5.2.52/240
4004_80D0	Software MUX Pad Control Register 52 (IOMUXC_PTC7)	32	R/W	0000_0060h	5.2.53/242
4004_80D4	Software MUX Pad Control Register 53 (IOMUXC_PTC8)	32	R/W	0000_0060h	5.2.54/244
4004_80D8	Software MUX Pad Control Register 54 (IOMUXC_PTC9)	32	R/W	0000_0060h	5.2.55/245
4004_80DC	Software MUX Pad Control Register 55 (IOMUXC_PTC10)	32	R/W	0000_0060h	5.2.56/247
4004_80E0	Software MUX Pad Control Register 56 (IOMUXC_PTC11)	32	R/W	0000_0060h	5.2.57/249
4004_80E4	Software MUX Pad Control Register 57 (IOMUXC_PTC12)	32	R/W	0000_0060h	5.2.58/251
4004_80E8	Software MUX Pad Control Register 58 (IOMUXC_PTC13)	32	R/W	0000_0060h	5.2.59/252
4004_80EC	Software MUX Pad Control Register 59 (IOMUXC_PTC14)	32	R/W	0000_0060h	5.2.60/254
4004_80F0	Software MUX Pad Control Register 60 (IOMUXC_PTC15)	32	R/W	0000_0060h	5.2.61/256
4004_80F4	Software MUX Pad Control Register 61 (IOMUXC_PTC16)	32	R/W	0000_0060h	5.2.62/258
4004_80F8	Software MUX Pad Control Register 62 (IOMUXC_PTC17)	32	R/W	0000_0060h	5.2.63/259
4004_80FC	Software MUX Pad Control Register 63 (IOMUXC_PTD31)	32	R/W	0000_0060h	5.2.64/261
4004_8100	Software MUX Pad Control Register 64 (IOMUXC_PTD30)	32	R/W	0000_0060h	5.2.65/263
4004_8104	Software MUX Pad Control Register 65 (IOMUXC_PTD29)	32	R/W	0000_0060h	5.2.66/264
4004_8108	Software MUX Pad Control Register 66 (IOMUXC_PTD28)	32	R/W	0000_0060h	5.2.67/266
4004_810C	Software MUX Pad Control Register 67 (IOMUXC_PTD27)	32	R/W	0000_0060h	5.2.68/268
4004_8110	Software MUX Pad Control Register 68 (IOMUXC_PTD26)	32	R/W	0000_0060h	5.2.69/270
4004_8114	Software MUX Pad Control Register 69 (IOMUXC_PTD25)	32	R/W	0000_0060h	5.2.70/271
4004_8118	Software MUX Pad Control Register 70 (IOMUXC_PTD24)	32	R/W	0000_0060h	5.2.71/273
4004_811C	Software MUX Pad Control Register 71 (IOMUXC_PTD23)	32	R/W	0000_0060h	5.2.72/275
4004_8120	Software MUX Pad Control Register 72 (IOMUXC_PTD22)	32	R/W	0000_0060h	5.2.73/277
4004_8124	Software MUX Pad Control Register 73 (IOMUXC_PTD21)	32	R/W	0000_0060h	5.2.74/279
4004_8128	Software MUX Pad Control Register 74 (IOMUXC_PTD20)	32	R/W	0000_0060h	5.2.75/280
4004_812C	Software MUX Pad Control Register 75 (IOMUXC_PTD19)	32	R/W	0000_0060h	5.2.76/282
4004_8130	Software MUX Pad Control Register 76 (IOMUXC_PTD18)	32	R/W	0000_0060h	5.2.77/284
4004_8134	Software MUX Pad Control Register 77 (IOMUXC_PTD17)	32	R/W	0000_0060h	5.2.78/286
4004_8138	Software MUX Pad Control Register 78 (IOMUXC_PTD16)	32	R/W	0000_0060h	5.2.79/287

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_813C	Software MUX Pad Control Register 79 (IOMUXC_PTD0)	32	R/W	0000_0060h	5.2.80/289
4004_8140	Software MUX Pad Control Register 80 (IOMUXC_PTD1)	32	R/W	0000_0060h	5.2.81/291
4004_8144	Software MUX Pad Control Register 81 (IOMUXC_PTD2)	32	R/W	0000_0060h	5.2.82/293
4004_8148	Software MUX Pad Control Register 82 (IOMUXC_PTD3)	32	R/W	0000_0060h	5.2.83/294
4004_814C	Software MUX Pad Control Register 83 (IOMUXC_PTD4)	32	R/W	0000_0060h	5.2.84/296
4004_8150	Software MUX Pad Control Register 84 (IOMUXC_PTD5)	32	R/W	0000_0060h	5.2.85/298
4004_8154	Software MUX Pad Control Register 85 (IOMUXC_PTD6)	32	R/W	0000_0060h	5.2.86/300
4004_8158	Software MUX Pad Control Register 86 (IOMUXC_PTD7)	32	R/W	0000_0060h	5.2.87/301
4004_815C	Software MUX Pad Control Register 87 (IOMUXC_PTD8)	32	R/W	0000_0060h	5.2.88/303
4004_8160	Software MUX Pad Control Register 88 (IOMUXC_PTD9)	32	R/W	0000_0060h	5.2.89/305
4004_8164	Software MUX Pad Control Register 89 (IOMUXC_PTD10)	32	R/W	0000_0060h	5.2.90/307
4004_8168	Software MUX Pad Control Register 90 (IOMUXC_PTD11)	32	R/W	0000_0060h	5.2.91/308
4004_816C	Software MUX Pad Control Register 91 (IOMUXC_PTD12)	32	R/W	0000_0060h	5.2.92/310
4004_8170	Software MUX Pad Control Register 92 (IOMUXC_PTD13)	32	R/W	0000_0060h	5.2.93/312
4004_8174	Software MUX Pad Control Register 93 (IOMUXC_PTB23)	32	R/W	0030_0061h	5.2.94/313
4004_8178	Software MUX Pad Control Register 94 (IOMUXC_PTB24)	32	R/W	0030_0061h	5.2.95/315
4004_817C	Software MUX Pad Control Register 95 (IOMUXC_PTB25)	32	R/W	0030_0061h	5.2.96/317
4004_8180	Software MUX Pad Control Register 96 (IOMUXC_PTB26)	32	R/W	0030_0061h	5.2.97/319
4004_8184	Software MUX Pad Control Register 97 (IOMUXC_PTB27)	32	R/W	0030_0061h	5.2.98/320
4004_8188	Software MUX Pad Control Register 98 (IOMUXC_PTB28)	32	R/W	0030_0061h	5.2.99/322
4004_818C	Software MUX Pad Control Register 99 (IOMUXC_PTC26)	32	R/W	0030_0061h	5.2.100/324
4004_8190	Software MUX Pad Control Register 100 (IOMUXC_PTC27)	32	R/W	0030_0061h	5.2.101/325
4004_8194	Software MUX Pad Control Register 101 (IOMUXC_PTC28)	32	R/W	0030_0061h	5.2.102/327
4004_8198	Software MUX Pad Control Register 102 (IOMUXC_PTC29)	32	R/W	0030_0061h	5.2.103/329
4004_819C	Software MUX Pad Control Register 103 (IOMUXC_PTC30)	32	R/W	0030_0061h	5.2.104/331
4004_81A0	Software MUX Pad Control Register 104 (IOMUXC_PTC31)	32	R/W	0030_0061h	5.2.105/332
4004_81A4	Software MUX Pad Control Register 105 (IOMUXC_PTE0)	32	R/W	0020_0044h	5.2.106/334
4004_81A8	Software MUX Pad Control Register 106 (IOMUXC_PTE1)	32	R/W	0020_0044h	5.2.107/336
4004_81AC	Software MUX Pad Control Register 107 (IOMUXC_PTE2)	32	R/W	0000_0060h	5.2.108/337
4004_81B0	Software MUX Pad Control Register 108 (IOMUXC_PTE3)	32	R/W	0000_0060h	5.2.109/339
4004_81B4	Software MUX Pad Control Register 109 (IOMUXC_PTE4)	32	R/W	0000_0060h	5.2.110/341
4004_81B8	Software MUX Pad Control Register 110 (IOMUXC_PTE5)	32	R/W	0000_0060h	5.2.111/342
4004_81BC	Software MUX Pad Control Register 111 (IOMUXC_PTE6)	32	R/W	0000_0060h	5.2.112/344
4004_81C0	Software MUX Pad Control Register 112 (IOMUXC_PTE7)	32	R/W	0030_0060h	5.2.113/346
4004_81C4	Software MUX Pad Control Register 113 (IOMUXC_PTE8)	32	R/W	0030_0060h	5.2.114/347
4004_81C8	Software MUX Pad Control Register 114 (IOMUXC_PTE9)	32	R/W	0030_0060h	5.2.115/349
4004_81CC	Software MUX Pad Control Register 115 (IOMUXC_PTE10)	32	R/W	0030_0060h	5.2.116/351
4004_81D0	Software MUX Pad Control Register 116 (IOMUXC_PTE11)	32	R/W	0030_0060h	5.2.117/352

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_81D4	Software MUX Pad Control Register 117 (IOMUXC_PTE12)	32	R/W	0030_0060h	5.2.118/354
4004_81D8	Software MUX Pad Control Register 118 (IOMUXC_PTE13)	32	R/W	0000_0060h	5.2.119/356
4004_81DC	Software MUX Pad Control Register 119 (IOMUXC_PTE14)	32	R/W	0000_0060h	5.2.120/357
4004_81E0	Software MUX Pad Control Register 120 (IOMUXC_PTE15)	32	R/W	0030_0060h	5.2.121/359
4004_81E4	Software MUX Pad Control Register 121 (IOMUXC_PTE16)	32	R/W	0030_0060h	5.2.122/361
4004_81E8	Software MUX Pad Control Register 122 (IOMUXC_PTE17)	32	R/W	0030_0060h	5.2.123/362
4004_81EC	Software MUX Pad Control Register 123 (IOMUXC_PTE18)	32	R/W	0030_0060h	5.2.124/364
4004_81F0	Software MUX Pad Control Register 124 (IOMUXC_PTE19)	32	R/W	0030_0060h	5.2.125/366
4004_81F4	Software MUX Pad Control Register 125 (IOMUXC_PTE20)	32	R/W	0030_0060h	5.2.126/367
4004_81F8	Software MUX Pad Control Register 126 (IOMUXC_PTE21)	32	R/W	0000_0060h	5.2.127/369
4004_81FC	Software MUX Pad Control Register 127 (IOMUXC_PTE22)	32	R/W	0000_0060h	5.2.128/371
4004_8200	Software MUX Pad Control Register 128 (IOMUXC_PTE23)	32	R/W	0030_0060h	5.2.129/372
4004_8204	Software MUX Pad Control Register 129 (IOMUXC_PTE24)	32	R/W	0030_0060h	5.2.130/374
4004_8208	Software MUX Pad Control Register 130 (IOMUXC_PTE25)	32	R/W	0030_0060h	5.2.131/376
4004_820C	Software MUX Pad Control Register 131 (IOMUXC_PTE26)	32	R/W	0030_0060h	5.2.132/377
4004_8210	Software MUX Pad Control Register 132 (IOMUXC_PTE27)	32	R/W	0030_0060h	5.2.133/379
4004_8214	Software MUX Pad Control Register 133 (IOMUXC_PTE28)	32	R/W	0030_0060h	5.2.134/381
4004_8218	Software MUX Pad Control Register 134 (IOMUXC_PTA7)	32	R/W	0000_0060h	5.2.135/383
4004_821C	Software MUX DDR RESET Pad Configuration Register (IOMUXC_DDR_RESETB)	32	R/W	0001_0060h	5.2.136/385
4004_8220	Software MUX DDR A15 Pad Control Register (IOMUXC_DDR_A_15)	32	R/W	0001_0060h	5.2.137/386
4004_8224	Software MUX DDR A14 Pad Control Register (IOMUXC_DDR_A_14)	32	R/W	0001_0060h	5.2.138/388
4004_8228	Software MUX DDR A13 Pad Control Register (IOMUXC_DDR_A_13)	32	R/W	0001_0060h	5.2.139/389
4004_822C	Software MUX DDR A12 Pad Control Register (IOMUXC_DDR_A_12)	32	R/W	0001_0060h	5.2.140/391
4004_8230	Software MUX DDR A11 Pad Control Register (IOMUXC_DDR_A_11)	32	R/W	0001_0060h	5.2.141/392
4004_8234	Software MUX DDR A10 Pad Control Register (IOMUXC_DDR_A_10)	32	R/W	0001_0060h	5.2.142/394
4004_8238	Software MUX DDR A9 Pad Control Register (IOMUXC_DDR_A_9)	32	R/W	0001_0060h	5.2.143/395
4004_823C	Software MUX DDR A8 Pad Control Register (IOMUXC_DDR_A_8)	32	R/W	0001_0060h	5.2.144/397
4004_8240	Software MUX DDR A7 Pad Control Register (IOMUXC_DDR_A_7)	32	R/W	0001_0060h	5.2.145/398
4004_8244	Software MUX DDR A6 Pad Control Register (IOMUXC_DDR_A_6)	32	R/W	0001_0060h	5.2.146/400

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_8248	Software MUX DDR A5 Pad Control Register (IOMUXC_DDR_A_5)	32	R/W	0001_0060h	5.2.147/401
4004_824C	Software MUX DDR A4 Pad Control Register (IOMUXC_DDR_A_4)	32	R/W	0001_0060h	5.2.148/403
4004_8250	Software MUX DDR Pad A3 Control Register (IOMUXC_DDR_A_3)	32	R/W	0001_0060h	5.2.149/404
4004_8254	Software MUX DDR A2 Pad Control Register (IOMUXC_DDR_A_2)	32	R/W	0001_0060h	5.2.150/406
4004_8258	Software MUX DDR A1 Pad Control Register (IOMUXC_DDR_A_1)	32	R/W	0001_0060h	5.2.151/407
4004_825C	Software MUX DDR A0 Pad Control Register (IOMUXC_DDR_A_0)	32	R/W	0001_0060h	5.2.152/409
4004_8260	Software MUX DDR BA2 Pad Control Register (IOMUXC_DDR_BA_2)	32	R/W	0001_0060h	5.2.153/410
4004_8264	Software MUX DDR BA1 Pad Control Register (IOMUXC_DDR_BA_1)	32	R/W	0001_0060h	5.2.154/412
4004_8268	Software MUX DDR BA0 Pad Control Register (IOMUXC_DDR_BA_0)	32	R/W	0001_0060h	5.2.155/413
4004_826C	Software MUX DDR CAS Pad Control Register (IOMUXC_DDR_CAS_B)	32	R/W	0001_0060h	5.2.156/415
4004_8270	Software MUX DDR CKE0 Pad Control Register (IOMUXC_DDR_CKE_0)	32	R/W	0001_0068h	5.2.157/416
4004_8274	Software MUX DDR CLK0 Pad Control Register (IOMUXC_DDR_CLK_0)	32	R/W	0001_0060h	5.2.158/418
4004_8278	Software MUX DDR CS B0 Pad Control Register (IOMUXC_DDR_CS_B_0)	32	R/W	0001_0060h	5.2.159/419
4004_827C	Software MUX DDR CS D15 Pad Control Register (IOMUXC_DDR_CS_D_15)	32	R/W	0001_0060h	5.2.160/421
4004_8280	Software MUX DDR CS D14 Pad Control Register (IOMUXC_DDR_CS_D_14)	32	R/W	0001_0060h	5.2.161/422
4004_8284	Software MUX DDR CS D13 Pad Control Register (IOMUXC_DDR_CS_D_13)	32	R/W	0001_0060h	5.2.162/424
4004_8288	Software MUX DDR CS D12 Pad Control Register (IOMUXC_DDR_CS_D_12)	32	R/W	0001_0060h	5.2.163/425
4004_828C	Software MUX DDR CS D11 Pad Control Register (IOMUXC_DDR_CS_D_11)	32	R/W	0001_0060h	5.2.164/427
4004_8290	Software MUX DDR CS D10 Pad Control Register (IOMUXC_DDR_CS_D_10)	32	R/W	0001_0060h	5.2.165/428
4004_8294	Software MUX DDR CS D9 Pad Control Register (IOMUXC_DDR_CS_D_9)	32	R/W	0001_0060h	5.2.166/430
4004_8298	Software MUX DDR CS D8 Pad Control Register (IOMUXC_DDR_CS_D_8)	32	R/W	0001_0060h	5.2.167/431
4004_829C	Software MUX DDR CS D7 Pad Control Register (IOMUXC_DDR_CS_D_7)	32	R/W	0001_0060h	5.2.168/433

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_82A0	Software MUX DDR CS D6 Pad Control Register (IOMUXC_DDR_CS_D_6)	32	R/W	0001_0060h	5.2.169/434
4004_82A4	Software MUX DDR CS D5 Pad Control Register (IOMUXC_DDR_CS_D_5)	32	R/W	0001_0060h	5.2.170/436
4004_82A8	Software MUX DDR CS D4 Pad Control Register (IOMUXC_DDR_CS_D_4)	32	R/W	0001_0060h	5.2.171/437
4004_82AC	Software MUX DDR CS D3 Pad Control Register (IOMUXC_DDR_CS_D_3)	32	R/W	0001_0060h	5.2.172/439
4004_82B0	Software MUX DDR CS D2 Pad Control Register (IOMUXC_DDR_CS_D_2)	32	R/W	0001_0060h	5.2.173/440
4004_82B4	Software MUX DDR CS D1 Pad Control Register (IOMUXC_DDR_CS_D_1)	32	R/W	0001_0060h	5.2.174/442
4004_82B8	Software MUX DDR CS D0 Pad Control Register (IOMUXC_DDR_CS_D_0)	32	R/W	0001_0060h	5.2.175/443
4004_82BC	Software MUX DDR DQM1 Pad Control Register (IOMUXC_DDR_DQM_1)	32	R/W	0001_0060h	5.2.176/445
4004_82C0	Software MUX DDR DQM0 Pad Control Register 0 (IOMUXC_DDR_DQM_0)	32	R/W	0001_0060h	5.2.177/446
4004_82C4	Software MUX DDR DQS1 Pad Control Register 1 (IOMUXC_DDR_DQS_1)	32	R/W	0001_0060h	5.2.178/448
4004_82C8	Software MUX DDR DQS0 Pad Control Register 0 (IOMUXC_DDR_DQS_0)	32	R/W	0001_0060h	5.2.179/449
4004_82CC	Software MUX DDR RAS Pad Control Register (IOMUXC_DDR_RAS_B)	32	R/W	0001_0060h	5.2.180/451
4004_82D0	Software MUX DDR WE Pad Control Register (IOMUXC_DDR_WE_B)	32	R/W	0001_0060h	5.2.181/452
4004_82D4	Software MUX DDR ODT0 Pad Control Register (IOMUXC_DDR_ODT_0)	32	R/W	0001_0060h	5.2.182/454
4004_82D8	Software MUX DDR ODT1 Pad Control Register (IOMUXC_DDR_ODT_1)	32	R/W	0001_0060h	5.2.183/455
4004_82DC	Software MUX Dummy DDRBYTE1 Pad Control Register (IOMUXC_DUMMY_DDRBYTE1)	32	R/W	0001_0060h	5.2.184/457
4004_82E0	Software MUX Dummy DDRBYTE2 Pad Control Register (IOMUXC_DUMMY_DDRBYTE2)	32	R/W	0001_0060h	5.2.185/458
4004_82EC	CCM Audio External Clock Input Select Register (IOMUXC_CCM_AUD_EXT_CLK_SELECT_INPUT)	32	R/W	0000_0000h	5.2.186/460
4004_82F0	CCM Ethernet External Clock Input Select Register (IOMUXC_CCM_ENET_EXT_CLK_SELECT_INPUT)	32	R/W	0000_0000h	5.2.187/460
4004_82F4	CCM Ethernet TS Clock Input Select Register (IOMUXC_CCM_ENET_TS_CLK_SELECT_INPUT)	32	R/W	0000_0000h	5.2.188/461
4004_82F8	DSPI1 SCK Input Select Register (IOMUXC_DSPI1_IPP_IND_SCK_SELECT_INPUT)	32	R/W	0000_0000h	5.2.189/462
4004_82FC	DSPI1 SIN Input Select Register (IOMUXC_DSPI1_IPP_IND_SIN_SELECT_INPUT)	32	R/W	0000_0000h	5.2.190/463

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_8300	DSPI1 SS Input Select Register (IOMUXC_DSPI1_IPP_IND_SS_B_SELECT_INPUT)	32	R/W	0000_0000h	5.2.191/464
4004_8304	Ethernet MAC0 TIMER0 Input Select Register (IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_0_SELECT_INPUT)	32	R/W	0000_0000h	5.2.192/465
4004_8308	Ethernet MAC0 TIMER1 Input Select Register (IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_1_SELECT_INPUT)	32	R/W	0000_0000h	5.2.193/466
4004_830C	ESAI FST Input Select Register (IOMUXC_ESAI_IPP_IND_FST_SELECT_INPUT)	32	R/W	0000_0000h	5.2.194/467
4004_8310	ESAI SCKT Input Select Register (IOMUXC_ESAI_IPP_IND_SCKT_SELECT_INPUT)	32	R/W	0000_0000h	5.2.195/468
4004_8314	ESAI SDO0 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO0_SELECT_INPUT)	32	R/W	0000_0000h	5.2.196/469
4004_8318	ESAI SDO1 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO1_SELECT_INPUT)	32	R/W	0000_0000h	5.2.197/470
4004_831C	ESAI SDO2 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO2_SDI3_SELECT_INPUT)	32	R/W	0000_0000h	5.2.198/471
4004_8320	ESAI SDO3 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO3_SDI2_SELECT_INPUT)	32	R/W	0000_0000h	5.2.199/472
4004_8324	ESAI SDO4 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO4_SDI1_SELECT_INPUT)	32	R/W	0000_0000h	5.2.200/473
4004_8328	ESAI SDO5 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO5_SDI0_SELECT_INPUT)	32	R/W	0000_0000h	5.2.201/474
4004_832C	FlexTimer1 CH0 Input Select Register (IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_0_SELECT_INPUT)	32	R/W	0000_0000h	5.2.202/475
4004_8330	FlexTimer1 CH1 Input Select Register (IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_1_SELECT_INPUT)	32	R/W	0000_0000h	5.2.203/476
4004_8334	FlexTimer1 PHA Input Select Register (IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHA_SELECT_INPUT)	32	R/W	0000_0000h	5.2.204/477
4004_8338	FlexTimer1 PHB Input Select Register (IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHB_SELECT_INPUT)	32	R/W	0000_0000h	5.2.205/478
4004_833C	I2C0 SCL Input Select Register (IOMUXC_I2C0_IPP_SCL_IND_SELECT_INPUT)	32	R/W	0000_0000h	5.2.206/478
4004_8340	I2C0 SDA Input Select Register (IOMUXC_I2C0_IPP_SDA_IND_SELECT_INPUT)	32	R/W	0000_0000h	5.2.207/479
4004_8344	I2C1 SCL Input Select Register (IOMUXC_I2C1_IPP_SCL_IND_SELECT_INPUT)	32	R/W	0000_0000h	5.2.208/480
4004_8348	I2C1 SDA Input Select Register (IOMUXC_I2C1_IPP_SDA_IND_SELECT_INPUT)	32	R/W	0000_0000h	5.2.209/480

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_834C	I2C2 SCL Input Select Register (IOMUXC_I2C2_IPP_SCL_IND_SELECT_INPUT)	32	R/W	0000_0000h	5.2.210/481
4004_8350	I2C2 SDA Input Select Register (IOMUXC_I2C2_IPP_SDA_IND_SELECT_INPUT)	32	R/W	0000_0000h	5.2.211/482
4004_8354	MediaLB Clock Input Select Register (IOMUXC_MLB_TOP_MLBCLK_IN_SELECT_INPUT)	32	R/W	0000_0000h	5.2.212/483
4004_8358	MediaLB Data Input Select Register (IOMUXC_MLB_TOP_MLBDAT_IN_SELECT_INPUT)	32	R/W	0000_0000h	5.2.213/484
4004_835C	MediaLB Signal Input Select Register (IOMUXC_MLB_TOP_MLBSIG_IN_SELECT_INPUT)	32	R/W	0000_0000h	5.2.214/485
4004_8360	SAI1 TXSYNC Input Select Register (IOMUXC_SAI1_IPP_IND_SAI_TXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	5.2.215/486
4004_8364	SAI2 RXBCLK Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_RXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	5.2.216/486
4004_8368	SAI2 RXDATA0 Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_RXDATA_0_SELECT_INPUT)	32	R/W	0000_0000h	5.2.217/487
4004_836C	SAI2 RXSYNC Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_RXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	5.2.218/488
4004_8370	SAI2 TXBLCK Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_TXBCLK_SELECT_INPUT)	32	R/W	0000_0000h	5.2.219/488
4004_8374	SAI2 TXSYNC Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_TXSYNC_SELECT_INPUT)	32	R/W	0000_0000h	5.2.220/489
4004_8378	UART FLX1 CTS Input Select Register (IOMUXC_SCI_FLX1_IPP_IND_CTS_B_SELECT_INPUT)	32	R/W	0000_0000h	5.2.221/490
4004_837C	UART FLX1 RX Input Select Register (IOMUXC_SCI_FLX1_IPP_IND_SCI_RX_SELECT_INPUT)	32	R/W	0000_0000h	5.2.222/490
4004_8380	UART FLX1 TX Input Select Register (IOMUXC_SCI_FLX1_IPP_IND_SCI_TX_SELECT_INPUT)	32	R/W	0000_0000h	5.2.223/491
4004_8384	UART FLX2 CTS Input Select Register (IOMUXC_SCI_FLX2_IPP_IND_CTS_B_SELECT_INPUT)	32	R/W	0000_0000h	5.2.224/492
4004_8388	UART FLX2 RX Input Select Register (IOMUXC_SCI_FLX2_IPP_IND_SCI_RX_SELECT_INPUT)	32	R/W	0000_0000h	5.2.225/492
4004_838C	UART FLX2 TX Input Select Register (IOMUXC_SCI_FLX2_IPP_IND_SCI_TX_SELECT_INPUT)	32	R/W	0000_0000h	5.2.226/493
4004_8390	UART FLX3 RX Input Select Register (IOMUXC_SCI_FLX3_IPP_IND_SCI_RX_SELECT_INPUT)	32	R/W	0000_0000h	5.2.227/494
4004_8394	UART FLX3 TX Input Select Register (IOMUXC_SCI_FLX3_IPP_IND_SCI_TX_SELECT_INPUT)	32	R/W	0000_0000h	5.2.228/495
4004_8398	BOOTCFG18 Input Select Register (IOMUXC_SRC_IPP_BOOT_CFG_18_SELECT_INPUT)	32	R/W	0000_0000h	5.2.229/496
4004_839C	BOOTCFG19 Input Select Register (IOMUXC_SRC_IPP_BOOT_CFG_19_SELECT_INPUT)	32	R/W	0000_0000h	5.2.230/497
4004_83A0	BOOTCFG20 Input Select Register (IOMUXC_SRC_IPP_BOOT_CFG_20_SELECT_INPUT)	32	R/W	0000_0000h	5.2.231/498

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_83A4	Video Decoder Input Select Register (IOMUXC_VIDEO_IN0_IPP_IND_DE_SELECT_INPUT)	32	R/W	0000_0000h	5.2.232/498
4004_83A8	Video IN0 Input Select Register (IOMUXC_VIDEO_IN0_IPP_IND_FID_SELECT_INPUT)	32	R/W	0000_0000h	5.2.233/499
4004_83AC	Video PIXCLK Input Select Register (IOMUXC_VIDEO_IN0_IPP_IND_PIX_CLK_SELECT_INPUT)	32	R/W	0000_0000h	5.2.234/500

5.2.1 Software MUX Pad Control Register 0 (IOMUXC_PTA6)

Address: 4004_8000h base + 0h offset = 4004_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA6 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTA6. **NOTE:** Pad PTA6 is involved in Daisy Chain. • Config CCM_ENET_EXT_CLK_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[0] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII_CLKOUT of instance: ccm. 010 Select mux mode: ALT2 mux port: RMII_CLKIN of instance: ccm. Used as MAC0-TXCLK when MAC0-MII is enabled. 100 Select mux mode: ALT4 mux port: TCON[11] of instance: tcon1. 111 Select mux mode: ALT7 mux port: DATA_OUT[20] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA6. 00 Low (50 MHz) 01 medium (100MHz) 10 medium (100MHz) 11 high (200MHz)

Table continues on the next page...

IOMUXC_PTA6 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA6. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA6. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA6. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA6. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA6. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA6. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA6. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA6. 0 Disabled 1 Enabled

5.2.2 Software MUX Pad Control Register 1 (IOMUXC_PTA8)

Address: 4004_8000h base + 4h offset = 4004_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1

IOMUXC_PTA8 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTA8. **NOTE:** Pad PTA8 is involved in Daisy Chain. • Config Register IOMUXC_MLB_TOP_MLBCLK_IN_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[1] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TCLK of instance: debug. 100 Select mux mode: ALT4 mux port: DATA_OUT[18] of instance: tcon0. 111 Select mux mode: ALT7 mux port: MLBCLK of instance: mlb_top.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA8. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA8. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA8. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA8. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA8. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTA8 field descriptions (continued)

Field	Description
	010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA8. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA8. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA8. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA8. 0 Disabled 1 Enabled

5.2.3 Software MUX Pad Control Register 2 (IOMUXC_PTA9)

Address: 4004_8000h base + 8h offset = 4004_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1

IOMUXC_PTA9 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTA9 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTA9. **NOTE:** Pad PTA9 is involved in Daisy Chain. • Config Register IOMUXC_CCM_ENET_EXT_CLK_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[2] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TDI of instance: debug. 010 Select mux mode: ALT2 mux port: RMII_CLKOUT of instance: ccm. 011 Select mux mode: ALT3 mux port: RMII_CLKIN of instance: ccm. Used as MAC0-TXCLK when MAC0-MII is enabled. 100 Select mux mode: ALT4 mux port: DATA_OUT[19] of instance: tcon0. 110 Select mux mode: ALT6 mux port: IPP_WDOG_CA5_CM4_B of instance: wdog_glue.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA9. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA9. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA9. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA9. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA9. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA9. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTA9 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA9. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA9. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA9. 0 Disabled 1 Enabled

5.2.4 Software MUX Pad Control Register 3 (IOMUXC_PTA10)

Address: 4004_8000h base + Ch offset = 4004_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA10 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTA10. **NOTE:** Pad PTA10 is involved in Daisy Chain. • Config Register IOMUXC_CCM_AUD_EXT_CLK_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_CCM_ENET_TS_CLK_SELECT_INPUT for mode ALT6. • Config Register IOMUXC_MLB_TOP_MLBSIG_IN_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[3] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TDO of instance: debug. 010 Select mux mode: ALT2 mux port: EXT_AUDIO_MCLK of instance: ccm. 100 Select mux mode: ALT4 mux port: DATA_OUT[10] of instance: tcon0. 110 Select mux mode: ALT6 mux port: ENET_TS_CLKIN of instance: ccm. 111 Select mux mode: ALT7 mux port: MLBSIGNAL of instance: mlb_top.

Table continues on the next page...

IOMUXC_PTA10 field descriptions (continued)

Field	Description
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA10. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA10. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA10. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA10. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA10. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA10. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA10. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA10. 0 Disabled 1 Enabled

Table continues on the next page...

IOMUXC_PTA10 field descriptions (continued)

Field	Description
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA10. 0 Disabled 1 Enabled

5.2.5 Software MUX Pad Control Register 4 (IOMUXC_PTA11)

Address: 4004_8000h base + 10h offset = 4004_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1

IOMUXC_PTA11 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTA11. **NOTE:** Pad PTA11 is involved in Daisy Chain. • Config Register IOMUXC_MLB_TOP_MLBDAT_IN_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[4] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TMS of instance: debug. 100 Select mux mode: ALT4 mux port: DATA_OUT[11] of instance: tcon0. 111 Select mux mode: ALT7 mux port: MLBDATA of instance: mlb_top.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA11. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA11. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA11. 0 Output is CMOS 1 Output is open drain

Table continues on the next page...

IOMUXC_PTA11 field descriptions (continued)

Field	Description
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA11. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA11. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA11. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA11. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA11. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA11. 0 Disabled 1 Enabled

5.2.6 Software MUX Pad Control Register 5 (IOMUXC_PTA12)

Address: 4004_8000h base + 14h offset = 4004_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA12 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTA12. **NOTE:** Pad PTA12 is involved in Daisy Chain. • Config Register IOMUXC_CCM_AUD_EXT_CLK_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_I2C0_IPP_SCL_IND_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[5] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACECK of instance: platform. 010 Select mux mode: ALT2 mux port: EXT_AUDIO_MCLK of instance: ccm. 110 Select mux mode: ALT6 mux port: DATA[13] of instance: video_in0. 111 Select mux mode: ALT7 mux port: SCL of instance: i2c0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA12. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA12. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA12. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA12. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA12.

Table continues on the next page...

IOMUXC_PTA12 field descriptions (continued)

Field	Description
	000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA12. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA12. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA12. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA12. 0 Disabled 1 Enabled

5.2.7 Software MUX Pad Control Register 6 (IOMUXC_PTA16)

Address: 4004_8000h base + 18h offset = 4004_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA16 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTA16. **NOTE:** Pad PTA16 is involved in Daisy Chain. • Config Register IOMUXC_I2C0_IPP_SDA_IND_SELECT_INPUT for mode ALT7. • Config Register IOMUXC_SAI2_IPP_IND_SAI_TXBCLK_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[6] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[0] of instance: platform. 010 Select mux mode: ALT2 mux port: VBUS_EN_OTG of instance: usb. 011 Select mux mode: ALT3 mux port: ADC1SE0 of instance: adc1_da. 100 Select mux mode: ALT4 mux port: LCD29 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: TX_BCLK of instance: sai2. 110 Select mux mode: ALT6 mux port: DATA[14] of instance: video_in0. 111 Select mux mode: ALT7 mux port: SDA of instance: i2c0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA16. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA16. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA16. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA16. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA16. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA16. 00 100 kOhm Pull Down

Table continues on the next page...

IOMUXC_PTA16 field descriptions (continued)

Field	Description
	01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA16. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA16. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA16. 0 Disabled 1 Enabled

5.2.8 Software MUX Pad Control Register 7 (IOMUXC_PTA17)

Address: 4004_8000h base + 1Ch offset = 4004_801Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA17 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTA17. **NOTE:** Pad PTA17 is involved in Daisy Chain. • Config Register IOMUXC_I2C1_IPP_SCL_IND_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[7] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[1] of instance: platform. 010 Select mux mode: ALT2 mux port: VBUS_OC_OTG of instance: usb. 011 Select mux mode: ALT3 mux port: ADC1SE1 of instance: adc1_da.

Table continues on the next page...

IOMUXC_PTA17 field descriptions (continued)

Field	Description
	100 Select mux mode: ALT4 mux port: LCD30 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: USB0_SOF_PULSE of instance: usb. 110 Select mux mode: ALT6 mux port: DATA[15] of instance: video_in0. 111 Select mux mode: ALT7 mux port: SCL of instance: i2c1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA17. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA17. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA17. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA17. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA17. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA17. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA17. 0 Keeper enable 1 Pull enable

Table continues on the next page...

IOMUXC_PTA17 field descriptions (continued)

Field	Description
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA17. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA17. 0 Disabled 1 Enabled

5.2.9 Software MUX Pad Control Register 8 (IOMUXC_PTA18)

Address: 4004_8000h base + 20h offset = 4004_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA18 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTA18. **NOTE:** Pad PTA18 is involved in Daisy Chain. • Config Register IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHA_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_I2C1_IPP_SDA_IND_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[8] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[2] of instance: platform. 010 Select mux mode: ALT2 mux port: ADC0SE0 of instance: adc0_da. 011 Select mux mode: ALT3 mux port: QD_PHA of instance: flextimer1. 100 Select mux mode: ALT4 mux port: LCD31 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: TX_DATA of instance: sai2. 110 Select mux mode: ALT6 mux port: DATA[16] of instance: video_in0. 111 Select mux mode: ALT7 mux port: SDA of instance: i2c1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA18. 00 Low (50 MHz) 01 Medium (100 MHz)

Table continues on the next page...

IOMUXC_PTA18 field descriptions (continued)

Field	Description
	10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA18. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA18. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA18. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA18. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA18. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA18. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA18. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA18. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA18. 0 Disabled 1 Enabled

5.2.10 Software MUX Pad Control Register 9 (IOMUXC_PTA19)

Address: 4004_8000h base + 24h offset = 4004_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA19 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTA19. **NOTE:** Pad PTA19 is involved in Daisy Chain. • Config Register IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHB_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_SAI2_IPP_IND_SAI_TXSYNC_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[9] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[3] of instance: platform. 010 Select mux mode: ALT2 mux port: ADC0SE1 of instance: adc0_da. 011 Select mux mode: ALT3 mux port: QD_PHB of instance: flextimer1. 100 Select mux mode: ALT4 mux port: LCD32 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: TX_SYNC of instance: sai2. 110 Select mux mode: ALT6 mux port: DATA[17] of instance: video_in0. 111 Select mux mode: ALT7 mux port: QSCK_A of instance: quadspi1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA19. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA19. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA19. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA19.

Table continues on the next page...

IOMUXC_PTA19 field descriptions (continued)

Field	Description
	0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA19. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA19. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA19. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA19. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA19. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA19. 0 Disabled 1 Enabled

5.2.11 Software MUX Pad Control Register 10 (IOMUXC_PTA20)

Address: 4004_8000h base + 28h offset = 4004_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA20 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTA20. **NOTE:** Pad PTA20 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX3_IPP_IND_SCI_TX_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: GPIO[10] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[4] of instance: platform. 100 Select mux mode: ALT4 mux port: LCD33 of instance: lcd_64f6b. 110 Select mux mode: ALT6 mux port: TX of instance: sci_flx3. 111 Select mux mode: ALT7 mux port: TCON[1] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA20. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA20. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA20. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA20. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA20. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA20. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTA20 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA20. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA20. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA20. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA20. 0 Disabled 1 Enabled

5.2.12 Software MUX Pad Control Register 11 (IOMUXC_PTA21)

Address: 4004_8000h base + 2Ch offset = 4004_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA21 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTA21. **NOTE:** Pad PTA21 is involved in Daisy Chain. • Config Register IOMUXC_SAI2_IPP_IND_SAI_RXBCLK_SELECT_INPUT for mode ALT5. • Config Register IOMUXC_SCI_FLX3_IPP_IND_SCI_RX_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: GPIO[11] of instance: rgpioc. Also, RXCLK for MAC0 is enabled in this mux mode so ensure obe is disabled if this pin is used for MAC0-MII instead of GPIO. 001 Select mux mode: ALT1 mux port: TRACED[5] of instance: platform. 101 Select mux mode: ALT5 mux port: RX_BCLK of instance: sai2. 110 Select mux mode: ALT6 mux port: RX of instance: sci_flx3. 111 Select mux mode: ALT7 mux port: TCON[2] of instance: tcon1.

Table continues on the next page...

IOMUXC_PTA21 field descriptions (continued)

Field	Description
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA21. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA21. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA21. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA21. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA21. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA21. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA21. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA21. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA21. 0 Disabled 1 Enabled

Table continues on the next page...

IOMUXC_PTA21 field descriptions (continued)

Field	Description
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA21. 0 Disabled 1 Enabled

5.2.13 Software MUX Pad Control Register 12 (IOMUXC_PTA22)

Address: 4004_8000h base + 30h offset = 4004_8030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA22 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTA22. **NOTE:** Pad PTA22 is involved in Daisy Chain. • Config Register IOMUXC_I2C2_IPP_SCL_IND_SELECT_INPUT for mode ALT6. • Config Register IOMUXC_SAI2_IPP_IND_SAI_RXDATA_0_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[12] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[6] of instance: platform. 101 Select mux mode: ALT5 mux port: RX_DATA of instance: sai2. 110 Select mux mode: ALT6 mux port: SCL of instance: i2c2. 111 Select mux mode: ALT7 mux port: TCON[0] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA22. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA22. 0 Slow Slew Rate 1 Fast Slew Rate

Table continues on the next page...

IOMUXC_PTA22 field descriptions (continued)

Field	Description
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA22. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA22. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA22. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA22. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA22. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA22. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA22. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA22. 0 Disabled 1 Enabled

5.2.14 Software MUX Pad Control Register 13 (IOMUXC_PTA23)

Address: 4004_8000h base + 34h offset = 4004_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA23 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTA23. **NOTE:** Pad PTA23 is involved in Daisy Chain. • Config Register IOMUXC_I2C2_IPP_SDA_IND_SELECT_INPUT for mode ALT6. • Config Register IOMUXC_SAI2_IPP_IND_SAI_RXSYNC_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[13] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[7] of instance: platform. 101 Select mux mode: ALT5 mux port: RX_SYNC of instance: sai2. 110 Select mux mode: ALT6 mux port: SDA of instance: i2c2. 111 Select mux mode: ALT7 mux port: TCON[3] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA23. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA23. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA23. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA23. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA23.

Table continues on the next page...

IOMUXC_PTA23 field descriptions (continued)

Field	Description
	000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA23. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA23. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA23. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA23. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA23. 0 Disabled 1 Enabled

5.2.15 Software MUX Pad Control Register 14 (IOMUXC_PTA24)

Address: 4004_8000h base + 38h offset = 4004_8038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA24 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTA24. 000 Select mux mode: ALT0 mux port: GPIO[14] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[8] of instance: platform. 010 Select mux mode: ALT2 mux port: VBUS_EN of instance: usb. 101 Select mux mode: ALT5 mux port: CLK of instance: esdhc1. 110 Select mux mode: ALT6 mux port: TCON[4] of instance: tcon1. 111 Select mux mode: ALT7 mux port: PAD_CTRL of instance: ddr_test_logic.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA24. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA24. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA24. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA24. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA24. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA24. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA24.

Table continues on the next page...

IOMUXC_PTA24 field descriptions (continued)

Field	Description
	0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA24. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA24. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA24. 0 Disabled 1 Enabled

5.2.16 Software MUX Pad Control Register 15 (IOMUXC_PTA25)

Address: 4004_8000h base + 3Ch offset = 4004_803Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA25 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTA25. 000 Select mux mode: ALT0 mux port: GPIO[15] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[9] of instance: platform. 010 Select mux mode: ALT2 mux port: VBUS_OC of instance: usb. 101 Select mux mode: ALT5 mux port: CMD of instance: esdhc1. 110 Select mux mode: ALT6 mux port: TCON[5] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA25. 00 Low (50 MHz) 01 Medium (100 MHz)

Table continues on the next page...

IOMUXC_PTA25 field descriptions (continued)

Field	Description
	10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA25. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA25. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA25. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA25. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA25. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA25. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA25. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA25. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA25. 0 Disabled 1 Enabled

5.2.17 Software MUX Pad Control Register 16 (IOMUXC_PTA26)

Address: 4004_8000h base + 40h offset = 4004_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA26 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTA26. 000 Select mux mode: ALT0 mux port: GPIO[16] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[10] of instance: platform. 010 Select mux mode: ALT2 mux port: TX_BCLK of instance: sai3. 101 Select mux mode: ALT5 mux port: DAT0 of instance: esdhc1. 110 Select mux mode: ALT6 mux port: TCON[6] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA26. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA26. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA26. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA26. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA26. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTA26 field descriptions (continued)

Field	Description
	011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA26. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA26. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA26. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA26. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA26. 0 Disabled 1 Enabled

5.2.18 Software MUX Pad Control Register 17 (IOMUXC_PTA27)

Address: 4004_8000h base + 44h offset = 4004_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE				Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA27 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTA27 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTA27. 000 Select mux mode: ALT0 mux port: GPIO[17] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[11] of instance: platform. 010 Select mux mode: ALT2 mux port: RX_BCLK of instance: sai3. 101 Select mux mode: ALT5 mux port: DAT1 of instance: esdhc1. 110 Select mux mode: ALT6 mux port: TCON[7] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA27. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA27. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA27. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA27. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA27. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA27. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA27. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled

Table continues on the next page...

IOMUXC_PTA27 field descriptions (continued)

Field	Description
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA27. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA27. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA27. 0 Disabled 1 Enabled

5.2.19 Software MUX Pad Control Register 18 (IOMUXC_PTA28)

Address: 4004_8000h base + 48h offset = 4004_8048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA28 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTA28. 000 Select mux mode: ALT0 mux port: GPIO[18] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[12] of instance: platform. 010 Select mux mode: ALT2 mux port: RX_DATA of instance: sai3. 011 Select mux mode: ALT3 mux port: MAC1_TMR0 of instance: enet_swiahb. 100 Select mux mode: ALT4 mux port: TX of instance: sci_flx4. 101 Select mux mode: ALT5 mux port: DAT2 of instance: esdhc1. 110 Select mux mode: ALT6 mux port: TCON[8] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA28. 00 Low (50 MHz) 01 Medium (100 MHz)

Table continues on the next page...

IOMUXC_PTA28 field descriptions (continued)

Field	Description
	10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA28. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA28. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA28. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA28. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA28. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA28. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA28. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA28. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA28. 0 Disabled 1 Enabled

5.2.20 Software MUX Pad Control Register 19 (IOMUXC_PTA29)

Address: 4004_8000h base + 4Ch offset = 4004_804Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA29 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTA29. 000 Select mux mode: ALT0 mux port: GPIO[19] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[13] of instance: platform. 010 Select mux mode: ALT2 mux port: TX_DATA of instance: sai3. 011 Select mux mode: ALT3 mux port: MAC1_TMR1 of instance: enet_swiahb. 100 Select mux mode: ALT4 mux port: RX of instance: sci_flx4. 101 Select mux mode: ALT5 mux port: DAT3 of instance: esdhc1. 110 Select mux mode: ALT6 mux port: TCON[9] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA29. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA29. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA29. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA29. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA29. 000 output driver disabled;

Table continues on the next page...

IOMUXC_PTA29 field descriptions (continued)

Field	Description
	001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA29. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA29. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA29. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA29. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA29. 0 Disabled 1 Enabled

5.2.21 Software MUX Pad Control Register 20 (IOMUXC_PTA30)

Address: 4004_8000h base + 50h offset = 4004_8050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA30 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTA30. **NOTE:** Pad PTA30 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX3_IPP_IND_SCI_TX_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[20] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[14] of instance: platform. 010 Select mux mode: ALT2 mux port: RX_SYNC of instance: sai3. 011 Select mux mode: ALT3 mux port: MAC1_TMR2 of instance: enet_swiahb. 100 Select mux mode: ALT4 mux port: RTS of instance: sci_flx4. 101 Select mux mode: ALT5 mux port: SCL of instance: i2c3. 111 Select mux mode: ALT7 mux port: TX of instance: sci_flx3.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA30. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA30. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA30. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA30. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA30. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA30. 00 100 kOhm Pull Down 01 47 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTA30 field descriptions (continued)

Field	Description
	10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA30. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA30. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA30. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA30. 0 Disabled 1 Enabled

5.2.22 Software MUX Pad Control Register 21 (IOMUXC_PTA31)

Address: 4004_8000h base + 54h offset = 4004_8054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA31 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTA31. **NOTE:** Pad PTA31 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX3_IPP_IND_SCI_RX_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[21] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TRACED[15] of instance: platform. 010 Select mux mode: ALT2 mux port: TX_SYNC of instance: sai3. 100 Select mux mode: ALT4 mux port: CTS of instance: sci_flx4.

Table continues on the next page...

IOMUXC_PTA31 field descriptions (continued)

Field	Description
	101 Select mux mode: ALT5 mux port: SDA of instance: i2c3. 111 Select mux mode: ALT7 mux port: RX of instance: sci_flx3.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA31. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA31. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA31. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA31. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA31. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA31. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA31. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA31. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA31.

Table continues on the next page...

IOMUXC_PTA31 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA31. 0 Disabled 1 Enabled

5.2.23 Software MUX Pad Control Register 22 (IOMUXC_PTB0)

Address: 4004_8000h base + 58h offset = 4004_8058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB0 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTB0. **NOTE:** Pad PTB0 is involved in Daisy Chain. • Config Register IOMUXC_SAI2_IPP_IND_SAI_RXBCLK_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[22] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[0] of instance: flextimer0. 010 Select mux mode: ALT2 mux port: ADC0SE2 of instance: adc0_da. 011 Select mux mode: ALT3 mux port: TRACECTL of instance: platform. 100 Select mux mode: ALT4 mux port: LCD34 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: RX_BCLK of instance: sai2. 110 Select mux mode: ALT6 mux port: DATA[18] of instance: video_in0. 111 Select mux mode: ALT7 mux port: QPCS0_A of instance: quadspi1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB0. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)

Table continues on the next page...

IOMUXC_PTB0 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB0. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB0. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB0. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB0. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB0. 0 Disabled 1 Enabled

5.2.24 Software MUX Pad Control Register 23 (IOMUXC_PTB1)

Address: 4004_8000h base + 5Ch offset = 4004_805Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB1 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTB1. **NOTE:** Pad PTB1 is involved in Daisy Chain. • Config Register IOMUXC_SAI2_IPP_IND_SAI_RXDATA_0_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[23] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[1] of instance: flextimer0. 010 Select mux mode: ALT2 mux port: ADC0SE3 of instance: adc0_da. 011 Select mux mode: ALT3 mux port: RCON30 of instance: src. 100 Select mux mode: ALT4 mux port: LCD35 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: RX_DATA of instance: sai2. 110 Select mux mode: ALT6 mux port: DATA[19] of instance: video_in0. 111 Select mux mode: ALT7 mux port: QSPI_IO3_A of instance: quadspi1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB1. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB1. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB1. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB1. 0 CMOS input 1 Schmitt trigger input

Table continues on the next page...

IOMUXC_PTB1 field descriptions (continued)

Field	Description
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB1. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB1. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB1. 0 Disabled 1 Enabled

5.2.25 Software MUX Pad Control Register 24 (IOMUXC_PTB2)

Address: 4004_8000h base + 60h offset = 4004_8060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB2 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTB2. **NOTE:** Pad PTB2 is involved in Daisy Chain. • Config Register IOMUXC_SAI2_IPP_IND_SAI_RXSYNC_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[24] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[2] of instance: flextimer0. 010 Select mux mode: ALT2 mux port: ADC1SE2 of instance: adc1_da. 011 Select mux mode: ALT3 mux port: RCON31 of instance: src. 100 Select mux mode: ALT4 mux port: LCD36 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: RX_SYNC of instance: sai2. 110 Select mux mode: ALT6 mux port: DATA[20] of instance: video_in0. 111 Select mux mode: ALT7 mux port: QSPI_IO2_A of instance: quadspi1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB2. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB2. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB2. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB2. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB2. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB2. 00 100 kOhm Pull Down 01 47 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTB2 field descriptions (continued)

Field	Description
	10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB2. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB2. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB2. 0 Disabled 1 Enabled

5.2.26 Software MUX Pad Control Register 25 (IOMUXC_PTB3)

Address: 4004_8000h base + 64h offset = 4004_8064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB3 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTB3. 000 Select mux mode: ALT0 mux port: GPIO[25] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[3] of instance: flextimer0. 010 Select mux mode: ALT2 mux port: ADC1SE3 of instance: adc1_da. 011 Select mux mode: ALT3 mux port: EXTRIG of instance: pdb. 100 Select mux mode: ALT4 mux port: LCD37 of instance: lcd_64f6b. 110 Select mux mode: ALT6 mux port: DATA[21] of instance: video_in0. 111 Select mux mode: ALT7 mux port: QSPI_IO1_A of instance: quadspi1.

Table continues on the next page...

IOMUXC_PTB3 field descriptions (continued)

Field	Description
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB3. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB3. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB3. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB3. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB3. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB3. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB3. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB3. 0 Disabled 1 Enabled

Table continues on the next page...

IOMUXC_PTB3 field descriptions (continued)

Field	Description
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB3. 0 Disabled 1 Enabled

5.2.27 Software MUX Pad Control Register 26 (IOMUXC_PTB4)

Address: 4004_8000h base + 68h offset = 4004_8068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB4 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTB4. **NOTE:** Pad PTB4 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX1_IPP_IND_SCI_TX_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_VIDEO_IN0_IPP_IND_FID_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[26] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[4] of instance: flextimer0. 010 Select mux mode: ALT2 mux port: TX of instance: sci_flx1. 011 Select mux mode: ALT3 mux port: ADC0SE4 of instance: adc0_da. 100 Select mux mode: ALT4 mux port: LCD38 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: VIU_FID of instance: video_in0. 110 Select mux mode: ALT6 mux port: DATA[22] of instance: video_in0. 111 Select mux mode: ALT7 mux port: QSPI_IO0_A of instance: quadspi1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB4. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB4.

Table continues on the next page...

IOMUXC_PTB4 field descriptions (continued)

Field	Description
	0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB4. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB4. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB4. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB4. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB4. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB4. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB4. 0 Disabled 1 Enabled

5.2.28 Software MUX Pad Control Register 27 (IOMUXC_PTB5)

Address: 4004_8000h base + 6Ch offset = 4004_806Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB5 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTB5. **NOTE:** Pad PTB5 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX1_IPP_IND_SCI_RX_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_VIDEO_IN0_IPP_IND_DE_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[27] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[5] of instance: flextimer0. 010 Select mux mode: ALT2 mux port: RX of instance: sci_flx1. 011 Select mux mode: ALT3 mux port: ADC1SE4 of instance: adc1_da. 100 Select mux mode: ALT4 mux port: LCD39 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: VIU_DE of instance: video_in0. 110 Reserved 111 Select mux mode: ALT7 mux port: DQS_A of instance: quadspi1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB5. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB5. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB5. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB5.

Table continues on the next page...

IOMUXC_PTB5 field descriptions (continued)

Field	Description
	0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB5. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB5. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB5. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB5. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB5. 0 Disabled 1 Enabled

5.2.29 Software MUX Pad Control Register 28 (IOMUXC_PTB6)

Address: 4004_8000h base + 70h offset = 4004_8070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB6 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTB6. **NOTE:** Pad PTB6 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX2_IPP_IND_SCI_TX_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[28] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[6] of instance: flextimer0. 010 Select mux mode: ALT2 mux port: RTS of instance: sci_flx1. 011 Select mux mode: ALT3 mux port: QPCS1_A of instance: quadspi0. 100 Select mux mode: ALT4 mux port: LCD40 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: FB_CLKOUT of instance: lpcg0. 110 Select mux mode: ALT6 mux port: HSYNC of instance: video_in0. 111 Select mux mode: ALT7 mux port: TX of instance: sci_flx2.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB6. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB6. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB6. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB6. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB6. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB6. 00 100 kOhm Pull Down 01 47 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTB6 field descriptions (continued)

Field	Description
	10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB6. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB6. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB6. 0 Disabled 1 Enabled

5.2.30 Software MUX Pad Control Register 29 (IOMUXC_PTB7)

Address: 4004_8000h base + 74h offset = 4004_8074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB7 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTB7. **NOTE:** Pad PTB7 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX1_IPP_IND_CTS_B_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_SCI_FLX2_IPP_IND_SCI_RX_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[29] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[7] of instance: flextimer0. 010 Select mux mode: ALT2 mux port: CTS of instance: sci_flx1. 011 Select mux mode: ALT3 mux port: QPCS1_B of instance: quadspi0. 100 Select mux mode: ALT4 mux port: LCD41 of instance: lcd_64f6b.

Table continues on the next page...

IOMUXC_PTB7 field descriptions (continued)

Field	Description
	110 Select mux mode: ALT6 mux port: VSYNC of instance: video_in0. 111 Select mux mode: ALT7 mux port: RX of instance: sci_flx2.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB7. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB7. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB7. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB7. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB7. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB7. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB7. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB7.

Table continues on the next page...

IOMUXC_PTB7 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB7. 0 Disabled 1 Enabled

5.2.31 Software MUX Pad Control Register 30 (IOMUXC_PTB8)

Address: 4004_8000h base + 78h offset = 4004_8078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB8 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTB8. **NOTE:** Pad PTB8 is involved in Daisy Chain. • Config Register IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_0_SELECT_INPUT for mode ALT1. • Config Register IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHA_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_VIDEO_IN0_IPP_IND_DE_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[30] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[0] of instance: flextimer1. 011 Select mux mode: ALT3 mux port: QD_PHA of instance: flextimer1. 101 Select mux mode: ALT5 mux port: VIU_DE of instance: video_in0. 111 Select mux mode: ALT7 mux port: DATA_OUT[24] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB8. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB8.

Table continues on the next page...

IOMUXC_PTB8 field descriptions (continued)

Field	Description
	0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB8. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB8. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB8. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB8. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB8. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB8. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB8. 0 Disabled 1 Enabled

5.2.32 Software MUX Pad Control Register 31 (IOMUXC_PTB9)

Address: 4004_8000h base + 7Ch offset = 4004_807Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB9 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTB9. **NOTE:** Pad PTB9 is involved in Daisy Chain. • Config Register IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_1_SELECT_INPUT for mode ALT1. • Config Register IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHB_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[31] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CH[1] of instance: flextimer1. 011 Select mux mode: ALT3 mux port: QD_PHB of instance: flextimer1. 111 Select mux mode: ALT7 mux port: DATA_OUT[25] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB9. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB9. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB9. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB9. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB9. 000 output driver disabled;

Table continues on the next page...

IOMUXC_PTB9 field descriptions (continued)

Field	Description
	001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB9. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB9. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB9. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB9. 0 Disabled 1 Enabled

5.2.33 Software MUX Pad Control Register 32 (IOMUXC_PTB10)

Address: 4004_8000h base + 80h offset = 4004_8080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB10 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTB10. **NOTE:** Pad PTB10 is involved in Daisy Chain. • Config Register IOMUXC_CCM_ENET_TS_CLK_SELECT_INPUT for mode ALT7. • Config Register IOMUXC_VIDEO_IN0_IPP_IND_DE_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[32] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TX of instance: sci_flx0. 100 Select mux mode: ALT4 mux port: TCON[4] of instance: tcon0. 101 Select mux mode: ALT5 mux port: VIU_DE of instance: video_in0. 110 CKO1 111 Select mux mode: ALT7 mux port: ENET_TS_CLKIN of instance: ccm.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB10. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB10. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB10. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB10. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB10. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB10. 00 100 kOhm Pull Down 01 47 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTB10 field descriptions (continued)

Field	Description
	10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB10. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB10. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB10. 0 Disabled 1 Enabled

5.2.34 Software MUX Pad Control Register 33 (IOMUXC_PTB11)

Address: 4004_8000h base + 84h offset = 4004_8084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB11 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTB11. **NOTE:** Pad PTB11 is involved in Daisy Chain. • Config Register IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_0_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[33] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RX of instance: sci_flx0. 100 Select mux mode: ALT4 mux port: TCON[5] of instance: tcon0. 101 Select mux mode: ALT5 mux port: SNVS_ALARM_OUT_B of instance: snvs_lp_wrapper.

Table continues on the next page...

IOMUXC_PTB11 field descriptions (continued)

Field	Description
	110 CKO2 111 Select mux mode: ALT7 mux port: MAC0_TMR0 of instance: enet_swiahb.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB11. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB11. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB11. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB11. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB11. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB11. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB11. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB11.

Table continues on the next page...

IOMUXC_PTB11 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB11. 0 Disabled 1 Enabled

5.2.35 Software MUX Pad Control Register 34 (IOMUXC_PTB12)

Address: 4004_8000h base + 88h offset = 4004_8088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB12 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTB12. **NOTE:** Pad PTB12 is involved in Daisy Chain. • Config Register IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_1_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[34] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RTS of instance: sci_flx0. 011 Select mux mode: ALT3 mux port: CS5 of instance: dspi0. 100 Select mux mode: ALT4 mux port: TCON[6] of instance: tcon0. 101 Select mux mode: ALT5 mux port: FB_AD[1] of instance: platform. 111 Select mux mode: ALT7 mux port: MAC0_TMR1 of instance: enet_swiahb.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB12. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB12.

Table continues on the next page...

IOMUXC_PTB12 field descriptions (continued)

Field	Description
	0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB12. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB12. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB12. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB12. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB12. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB12. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB12. 0 Disabled 1 Enabled

5.2.36 Software MUX Pad Control Register 35 (IOMUXC_PTB13)

Address: 4004_8000h base + 8Ch offset = 4004_808Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB13 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTB13. 000 Select mux mode: ALT0 mux port: GPIO[35] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CTS of instance: sci_flx0. 011 Select mux mode: ALT3 mux port: CS4 of instance: dspi0. 100 Select mux mode: ALT4 mux port: TCON[7] of instance: tcon0. 101 Select mux mode: ALT5 mux port: FB_AD[0] of instance: platform. 110 Select mux mode: ALT6 mux port: TRACECTL of instance: platform.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB13. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB13. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB13. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB13. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB13. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTB13 field descriptions (continued)

Field	Description
	010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB13. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB13. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB13. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB13. 0 Disabled 1 Enabled

5.2.37 Software MUX Pad Control Register 36 (IOMUXC_PTB14)

Address: 4004_8000h base + 90h offset = 4004_8090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB14 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTB14 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTB14. **NOTE:** Pad PTB14 is involved in Daisy Chain. • Config Register IOMUXC_I2C0_IPP_SCL_IND_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[36] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RXD of instance: can0. 010 Select mux mode: ALT2 mux port: SCL of instance: i2c0. 100 Select mux mode: ALT4 mux port: TCON[8] of instance: tcon0. 111 Select mux mode: ALT7 mux port: DATA_OUT[1] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB14. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB14. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB14. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB14. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB14. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB14. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB14.

Table continues on the next page...

IOMUXC_PTB14 field descriptions (continued)

Field	Description
	0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB14. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB14. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB14. 0 Disabled 1 Enabled

5.2.38 Software MUX Pad Control Register 37 (IOMUXC_PTB15)

Address: 4004_8000h base + 94h offset = 4004_8094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB15 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTB15. **NOTE:** Pad PTB15 is involved in Daisy Chain. • Config Register IOMUXC_I2C0_IPP_SDA_IND_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_VIDEO_IN0_IPP_IND_PIX_CLK_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[37] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TXD of instance: can0. 010 Select mux mode: ALT2 mux port: SDA of instance: i2c0. 100 Select mux mode: ALT4 mux port: TCON[9] of instance: tcon0. 111 Select mux mode: ALT7 mux port: PIX_CLK of instance: video_in0.
19–14 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTB15 field descriptions (continued)

Field	Description
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB15. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB15. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB15. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB15. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB15. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB15. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB15. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB15. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB15.

Table continues on the next page...

IOMUXC_PTB15 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled

5.2.39 Software MUX Pad Control Register 38 (IOMUXC_PTB16)

Address: 4004_8000h base + 98h offset = 4004_8098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB16 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTB16. **NOTE:** Pad PTB16 is involved in Daisy Chain. • Config Register IOMUXC_I2C1_IPP_SCL_IND_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[38] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RXD of instance: can1. 010 Select mux mode: ALT2 mux port: SCL of instance: i2c1. 100 Select mux mode: ALT4 mux port: TCON[10] of instance: tcon0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB16. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB16. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB16. 0 Output is CMOS 1 Output is open drain

Table continues on the next page...

IOMUXC_PTB16 field descriptions (continued)

Field	Description
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB16. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB16. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB16. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB16. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB16. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB16. 0 Disabled 1 Enabled

5.2.40 Software MUX Pad Control Register 39 (IOMUXC_PTB17)

Address: 4004_8000h base + 9Ch offset = 4004_809Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB17 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTB17. **NOTE:** Pad PTB17 is involved in Daisy Chain. • Config Register IOMUXC_I2C1_IPP_SDA_IND_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[39] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TXD of instance: can1. 010 Select mux mode: ALT2 mux port: SDA of instance: i2c1. 100 Select mux mode: ALT4 mux port: TCON[11] of instance: tcon0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB17. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB17. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB17. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB17. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB17. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTB17 field descriptions (continued)

Field	Description
	010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB17. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB17. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB17. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB17. 0 Disabled 1 Enabled

5.2.41 Software MUX Pad Control Register 40 (IOMUXC_PTB18)

Address: 4004_8000h base + A0h offset = 4004_80A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTB18 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTB18 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTB18. **NOTE:** Pad PTB18 is involved in Daisy Chain. • Config Register IOMUXC_CCM_AUD_EXT_CLK_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[40] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CS1 of instance: dspi0. 010 Select mux mode: ALT2 mux port: EXT_AUDIO_MCLK of instance: ccm. 100 CKO1 110 Select mux mode: ALT6 mux port: DATA[9] of instance: video_in0. 111 Reserved
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB18. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB18. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB18. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB18. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB18. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB18. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB18.

Table continues on the next page...

IOMUXC_PTB18 field descriptions (continued)

Field	Description
	0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB18. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB18. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB18. 0 Disabled 1 Enabled

5.2.42 Software MUX Pad Control Register 41 (IOMUXC_PTB19)

Address: 4004_8000h base + A4h offset = 4004_80A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTB19 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTB19. 000 Select mux mode: ALT0 mux port: GPIO[41] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: CS0 of instance: dspi0. 110 Select mux mode: ALT6 mux port: DATA[10] of instance: video_in0. 111 Reserved
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB19. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)

Table continues on the next page...

IOMUXC_PTB19 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB19. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB19. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB19. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB19. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB19. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB19. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB19. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB19. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB19. 0 Disabled 1 Enabled

5.2.43 Software MUX Pad Control Register 42 (IOMUXC_PTB20)

Address: 4004_8000h base + A8h offset = 4004_80A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB20 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTB20. 000 Select mux mode: ALT0 mux port: GPIO[42] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: SIN of instance: dspi0. 100 Select mux mode: ALT4 mux port: LCD42 of instance: lcd_64f6b. 110 Select mux mode: ALT6 mux port: DATA[11] of instance: video_in0. 111 Reserved
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB20. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB20. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB20. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB20. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB20. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTB20 field descriptions (continued)

Field	Description
	011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB20. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB20. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB20. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB20. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB20. 0 Disabled 1 Enabled

5.2.44 Software MUX Pad Control Register 43 (IOMUXC_PTB21)

Address: 4004_8000h base + ACh offset = 4004_80ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTB21 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTB21 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTB21. 000 Select mux mode: ALT0 mux port: GPIO[43] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: SOUT of instance: dspi0. 100 Select mux mode: ALT4 mux port: LCD43 of instance: lcd_64f6b. 110 Select mux mode: ALT6 mux port: DATA[12] of instance: video_in0. 111 Select mux mode: ALT7 mux port: DATA_OUT[1] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB21. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB21. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB21. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB21. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB21. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB21. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB21. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled

Table continues on the next page...

IOMUXC_PTB21 field descriptions (continued)

Field	Description
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB21. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB21. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB21. 0 Disabled 1 Enabled

5.2.45 Software MUX Pad Control Register 44 (IOMUXC_PTB22)

Address: 4004_8000h base + B0h offset = 4004_80B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTB22 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 3 IOMUX modes to be used for pad: PTB22. **NOTE:** Pad PTB22 is involved in Daisy Chain. • Config Register IOMUXC_VIDEO_IN0_IPP_IND_FID_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[44] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: SCK of instance: dspi0. 101 Select mux mode: ALT5 mux port: VIU_FID of instance: video_in0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB22. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)

Table continues on the next page...

IOMUXC_PTB22 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB22. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB22. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB22. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB22. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB22. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB22. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB22. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB22. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB22. 0 Disabled 1 Enabled

5.2.46 Software MUX Pad Control Register 45 (IOMUXC_PTC0)

Address: 4004_8000h base + B4h offset = 4004_80B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTC0 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTC0. **NOTE:** Pad PTC0 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SCKT_SELECT_INPUT for mode ALT4. • Config Register IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_0_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_SRC_IPP_BOOT_CFG_18_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[45] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII0_MDC/MII0_MDC of instance: enet_swiahb. 010 Select mux mode: ALT2 mux port: CH[0] of instance: flextimer1. 011 Select mux mode: ALT3 mux port: CS3 of instance: dspi0. 100 Select mux mode: ALT4 mux port: SCKT of instance: esai. 101 Select mux mode: ALT5 mux port: CLK of instance: esdhc0. 110 Select mux mode: ALT6 mux port: DATA[0] of instance: video_in0. 111 Select mux mode: ALT7 mux port: RCON18 of instance: src.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC0. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC0. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC0. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC0.

Table continues on the next page...

IOMUXC_PTC0 field descriptions (continued)

Field	Description
	0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC0. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC0. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC0. 0 Disabled 1 Enabled

5.2.47 Software MUX Pad Control Register 46 (IOMUXC_PTC1)

Address: 4004_8000h base + B8h offset = 4004_80B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTC1 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTC1. **NOTE:** Pad PTC1 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_FST_SELECT_INPUT for mode ALT4. • Config Register IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_1_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_SRC_IPP_BOOT_CFG_19_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[46] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII0_MDIO/MII0_MDIO of instance: enet_swiahb. 010 Select mux mode: ALT2 mux port: CH[1] of instance: flextimer1. 011 Select mux mode: ALT3 mux port: CS2 of instance: dspi0. 100 Select mux mode: ALT4 mux port: FST of instance: esai. 101 Select mux mode: ALT5 mux port: CMD of instance: esdhc0. 110 Select mux mode: ALT6 mux port: DATA[1] of instance: video_in0. 111 Select mux mode: ALT7 mux port: RCON19 of instance: src.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC1. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC1. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC1. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC1.

Table continues on the next page...

IOMUXC_PTC1 field descriptions (continued)

Field	Description
	00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC1. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC1. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC1. 0 Disabled 1 Enabled

5.2.48 Software MUX Pad Control Register 47 (IOMUXC_PTC2)

Address: 4004_8000h base + BCh offset = 4004_80BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTC2 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTC2. **NOTE:** Pad PTC2 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO0_SELECT_INPUT for mode ALT4. • Config Register IOMUXC_SCI_FLX1_IPP_IND_SCI_TX_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_SRC_IPP_BOOT_CFG_20_SELECT_INPUT for mode ALT7. 000 Select mux mode: ALT0 mux port: GPIO[47] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII0_RX_EN of instance: enet_swiahb.

Table continues on the next page...

IOMUXC_PTC2 field descriptions (continued)

Field	Description
	010 Select mux mode: ALT2 mux port: TX of instance: sci_flx1. 100 Select mux mode: ALT4 mux port: SDO0 of instance: esai. 101 Select mux mode: ALT5 mux port: DAT0 of instance: esdhc0. 110 Select mux mode: ALT6 mux port: DATA[2] of instance: video_in0. 111 Select mux mode: ALT7 mux port: RCON20 of instance: src.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC2. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC2. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC2. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC2. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC2. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC2. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC2.

Table continues on the next page...

IOMUXC_PTC2 field descriptions (continued)

Field	Description
	0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC2. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC2. 0 Disabled 1 Enabled

5.2.49 Software MUX Pad Control Register 48 (IOMUXC_PTC3)

Address: 4004_8000h base + C0h offset = 4004_80C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC3 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTC3. **NOTE:** Pad PTC3 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO1_SELECT_INPUT for mode ALT4. • Config Register IOMUXC_SCI_FLX1_IPP_IND_SCI_RX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[48] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII0_RXD[1]//MII0_RXD[1] of instance: enet_swiahb. 010 Select mux mode: ALT2 mux port: RX of instance: sci_flx1. 100 Select mux mode: ALT4 mux port: SDO1 of instance: esai. 101 Select mux mode: ALT5 mux port: DAT1 of instance: esdhc0. 110 Select mux mode: ALT6 mux port: DATA[3] of instance: video_in0. 111 Select mux mode: ALT7 mux port: DATA_OUT[18] of instance: tcon0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC3. 00 Low (50 MHz)

Table continues on the next page...

IOMUXC_PTC3 field descriptions (continued)

Field	Description
	01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC3. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC3. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC3. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC3. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC3. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC3. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC3. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC3. 0 Disabled 1 Enabled

5.2.50 Software MUX Pad Control Register 49 (IOMUXC_PTC4)

Address: 4004_8000h base + C4h offset = 4004_80C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC4 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTC4. **NOTE:** Pad PTC4 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO2_SDI3_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: GPIO[49] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII0_RXD[0]/MII0_RXD[0] of instance: enet_swiahb. 010 Select mux mode: ALT2 mux port: RTS of instance: sci_flx1. 011 Select mux mode: ALT3 mux port: CS1 of instance: dspi1. 100 Select mux mode: ALT4 mux port: SDO2 of instance: esai. 101 Select mux mode: ALT5 mux port: DAT2 of instance: esdhc0. 110 Select mux mode: ALT6 mux port: DATA[4] of instance: video_in0. 111 Select mux mode: ALT7 mux port: DATA_OUT[19] of instance: tcon0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC4. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC4. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC4. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC4. 0 CMOS input 1 Schmitt trigger input

Table continues on the next page...

IOMUXC_PTC4 field descriptions (continued)

Field	Description
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC4. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC4. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC4. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC4. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC4. 0 Disabled 1 Enabled

5.2.51 Software MUX Pad Control Register 50 (IOMUXC_PTC5)

Address: 4004_8000h base + C8h offset = 4004_80C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC5 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTC5. **NOTE:** Pad PTC5 is involved in Daisy Chain. • Config Register IOMUXC_DSPI1_IPP_IND_SS_B_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_ESAI_IPP_IND_SDO3_SDI2_SELECT_INPUT for mode ALT4. • Config Register IOMUXC_SCI_FLX1_IPP_IND_CTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[50] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII0_RXER/MII0_RXER of instance: enet_swiahb. 010 Select mux mode: ALT2 mux port: CTS of instance: sci_flx1. 011 Select mux mode: ALT3 mux port: CS0 of instance: dspi1. 100 Select mux mode: ALT4 mux port: SDO3 of instance: esai. 101 Select mux mode: ALT5 mux port: DAT3 of instance: esdhc0. 110 Select mux mode: ALT6 mux port: DATA[5] of instance: video_in0. 111 Select mux mode: ALT7 mux port: DATA_OUT[10] of instance: tcon0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC5. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC5. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC5. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC5. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC5. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC5.

Table continues on the next page...

IOMUXC_PTC5 field descriptions (continued)

Field	Description
	00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC5. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC5. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC5. 0 Disabled 1 Enabled

5.2.52 Software MUX Pad Control Register 51 (IOMUXC_PTC6)

Address: 4004_8000h base + CCh offset = 4004_80CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC6 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTC6. **NOTE:** Pad PTC6 is involved in Daisy Chain. • Config Register IOMUXC_DSPI1_IPP_IND_SIN_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_ESAI_IPP_IND_SDO5_SDI0_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: GPIO[51] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII0_TXD[1]/MII0_TXD[1] of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SIN of instance: dspi1.

Table continues on the next page...

IOMUXC_PTC6 field descriptions (continued)

Field	Description
	100 Select mux mode: ALT4 mux port: SDI0 of instance: esai. 101 Select mux mode: ALT5 mux port: WP of instance: esdhc0. 110 Select mux mode: ALT6 mux port: DATA[6] of instance: video_in0. 111 Select mux mode: ALT7 mux port: DATA_OUT[11] of instance: tcon0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC6. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC6. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC6. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC6. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC6. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC6. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC6. 0 Keeper enable 1 Pull enable

Table continues on the next page...

IOMUXC_PTC6 field descriptions (continued)

Field	Description
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC6. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC6. 0 Disabled 1 Enabled

5.2.53 Software MUX Pad Control Register 52 (IOMUXC_PTC7)

Address: 4004_8000h base + D0h offset = 4004_80D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC7 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTC7. **NOTE:** Pad PTC7 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO4_SDI1_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: GPIO[52] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII0_TXD[0]/MII0_TXD[0] of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SOUT of instance: dspi1. 100 Select mux mode: ALT4 mux port: SDI1 of instance: esai. 110 Select mux mode: ALT6 mux port: DATA[7] of instance: video_in0. 111 Select mux mode: ALT7 mux port: DATA_OUT[2] of instance: tcon0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC7. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)

Table continues on the next page...

IOMUXC_PTC7 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC7. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC7. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC7. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC7. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC7. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC7. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC7. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC7. 0 Disabled 1 Enabled

5.2.54 Software MUX Pad Control Register 53 (IOMUXC_PTC8)

Address: 4004_8000h base + D4h offset = 4004_80D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC8 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTC8. **NOTE:** Pad PTC8 is involved in Daisy Chain. • Config Register IOMUXC_DSPI1_IPP_IND_SCK_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[53] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII0_TXEN/MII0_TXEN of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SCK of instance: dspi1. 110 Select mux mode: ALT6 mux port: DATA[8] of instance: video_in0. 111 Select mux mode: ALT7 mux port: DATA_OUT[3] of instance: tcon0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC8. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC8. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC8. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC8. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC8. 000 output driver disabled;

Table continues on the next page...

IOMUXC_PTC8 field descriptions (continued)

Field	Description
	001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC8. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC8. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC8. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC8. 0 Disabled 1 Enabled

5.2.55 Software MUX Pad Control Register 54 (IOMUXC_PTC9)

Address: 4004_8000h base + D8h offset = 4004_80D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC9 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTC9. **NOTE:** Pad PTC9 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SCKT_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_MLB_TOP_MLBCLK_IN_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: GPIO[54] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII1_MDC of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SCKT of instance: esai. 110 Select mux mode: ALT6 mux port: MLBCLK of instance: mlb_top. 111 Select mux mode: ALT7 mux port: debug_out[0] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC9. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC9. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC9. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC9. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC9. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC9. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTC9 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC9. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC9. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC9. 0 Disabled 1 Enabled

5.2.56 Software MUX Pad Control Register 55 (IOMUXC_PTC10)

Address: 4004_8000h base + DCh offset = 4004_80DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC10 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTC10. **NOTE:** Pad PTC10 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_FST_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_MLB_TOP_MLBSIG_IN_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: GPIO[55] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII1_MDIO of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: FST of instance: esai. 110 Select mux mode: ALT6 mux port: MLBSIGNAL of instance: mlb_top. 111 Select mux mode: ALT7 mux port: debug_out[1] of instance: viu_mux.
19–14 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTC10 field descriptions (continued)

Field	Description
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC10. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC10. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC10. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC10. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC10. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC10. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC10. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC10. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC10.

Table continues on the next page...

IOMUXC_PTC10 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled

5.2.57 Software MUX Pad Control Register 56 (IOMUXC_PTC11)

Address: 4004_8000h base + E0h offset = 4004_80E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC11 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTC11. **NOTE:** Pad PTC11 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO0_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_MLB_TOP_MLBDAT_IN_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: GPIO[56] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII1_CRS_DV of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SDO0 of instance: esai. 110 Select mux mode: ALT6 mux port: MLBDATA of instance: mlb_top. 111 Select mux mode: ALT7 mux port: debug_out[2] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC11. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC11. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC11.

Table continues on the next page...

IOMUXC_PTC11 field descriptions (continued)

Field	Description
	0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC11. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC11. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC11. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC11. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC11. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC11. 0 Disabled 1 Enabled

5.2.58 Software MUX Pad Control Register 57 (IOMUXC_PTC12)

Address: 4004_8000h base + E4h offset = 4004_80E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC12 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTC12. **NOTE:** Pad PTC12 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO1_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_SAI2_IPP_IND_SAI_TXBCLK_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[57] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII1_RXD[1] of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SDO1 of instance: esai. 101 Select mux mode: ALT5 mux port: TX_BCLK of instance: sai2. 111 Select mux mode: ALT7 mux port: debug_out[3] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC12. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC12. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC12. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC12. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC12.

Table continues on the next page...

IOMUXC_PTC12 field descriptions (continued)

Field	Description
	000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC12. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC12. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC12. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC12. 0 Disabled 1 Enabled

5.2.59 Software MUX Pad Control Register 58 (IOMUXC_PTC13)

Address: 4004_8000h base + E8h offset = 4004_80E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC13 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTC13. **NOTE:** Pad PTC13 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO2_SDI3_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_SAI2_IPP_IND_SAI_RXBCLK_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[58] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII1_RXD[0] of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SDO2 of instance: esai. 101 Select mux mode: ALT5 mux port: RX_BCLK of instance: sai2. 111 Select mux mode: ALT7 mux port: debug_out[4] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC13. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC13. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC13. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC13. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC13. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC13. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTC13 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC13. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC13. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC13. 0 Disabled 1 Enabled

5.2.60 Software MUX Pad Control Register 59 (IOMUXC_PTC14)

Address: 4004_8000h base + ECh offset = 4004_80ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC14 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTC14. **NOTE:** Pad PTC14 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO3_SDI2_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_SAI2_IPP_IND_SAI_RXDATA_0_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[59] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII1_RXER of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SDO3 of instance: esai. 100 Select mux mode: ALT4 mux port: TX of instance: sci_flx5. 101 Select mux mode: ALT5 mux port: RX_DATA of instance: sai2. 110 Select mux mode: ALT6 mux port: ADC0SE6 of instance: adc0_da. 111 Select mux mode: ALT7 mux port: debug_out[5] of instance: viu_mux.

Table continues on the next page...

IOMUXC_PTC14 field descriptions (continued)

Field	Description
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC14. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC14. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC14. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC14. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC14. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC14. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC14. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC14. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC14. 0 Disabled 1 Enabled

Table continues on the next page...

IOMUXC_PTC14 field descriptions (continued)

Field	Description
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC14. 0 Disabled 1 Enabled

5.2.61 Software MUX Pad Control Register 60 (IOMUXC_PTC15)

Address: 4004_8000h base + F0h offset = 4004_80F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC15 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTC15. **NOTE:** Pad PTC15 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO5_SDI0_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[60] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII1_TXD[1] of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SDI0 of instance: esai. 100 Select mux mode: ALT4 mux port: RX of instance: sci_flx5. 101 Select mux mode: ALT5 mux port: TX_DATA of instance: sai2. 110 Select mux mode: ALT6 mux port: ADC0SE7 of instance: adc0_da. 111 Select mux mode: ALT7 mux port: debug_out[6] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC15. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC15. 0 Slow Slew Rate 1 Fast Slew Rate

Table continues on the next page...

IOMUXC_PTC15 field descriptions (continued)

Field	Description
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC15. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC15. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC15. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC15. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC15. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC15. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC15. 0 Disabled 1 Enabled

5.2.62 Software MUX Pad Control Register 61 (IOMUXC_PTC16)

Address: 4004_8000h base + F4h offset = 4004_80F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC16 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTC16. **NOTE:** Pad PTC16 is involved in Daisy Chain. • Config Register IOMUXC_ESAI_IPP_IND_SDO4_SDI1_SELECT_INPUT for mode ALT3. • Config Register IOMUXC_SAI2_IPP_IND_SAI_RXSYNC_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[61] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII1_TXD[0] of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: SDI1 of instance: esai. 100 Select mux mode: ALT4 mux port: RTS of instance: sci_flx5. 101 Select mux mode: ALT5 mux port: RX_SYNC of instance: sai2. 110 Select mux mode: ALT6 mux port: ADC1SE6 of instance: adc1_da. 111 Select mux mode: ALT7 mux port: debug_out[7] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC16. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC16. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC16. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC16. 0 CMOS input 1 Schmitt trigger input

Table continues on the next page...

IOMUXC_PTC16 field descriptions (continued)

Field	Description
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC16. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC16. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC16. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC16. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC16. 0 Disabled 1 Enabled

5.2.63 Software MUX Pad Control Register 62 (IOMUXC_PTC17)

Address: 4004_8000h base + F8h offset = 4004_80F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTC17 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTC17. **NOTE:** Pad PTC17 is involved in Daisy Chain. • Config Register IOMUXC_SAI2_IPP_IND_SAI_TXSYNC_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[62] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RMII1_TXEN of instance: enet_swiahb. 011 Select mux mode: ALT3 mux port: ADC1SE7 of instance: adc1_da. 100 Select mux mode: ALT4 mux port: CTS of instance: sci_flx5. 101 Select mux mode: ALT5 mux port: TX_SYNC of instance: sai2. 110 Select mux mode: ALT6 mux port: USB1_SOF_PULSE of instance: usb. 111 Select mux mode: ALT7 mux port: debug_out[8] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC17. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC17. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC17. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC17. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC17. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC17. 00 100 kOhm Pull Down 01 47 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTC17 field descriptions (continued)

Field	Description
	10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC17. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC17. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC17. 0 Disabled 1 Enabled

5.2.64 Software MUX Pad Control Register 63 (IOMUXC_PTD31)

Address: 4004_8000h base + FCh offset = 4004_80FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD31 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD31. 000 Select mux mode: ALT0 mux port: GPIO[63] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[31] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[15] of instance: nfc_mlc. 100 Select mux mode: ALT4 mux port: CH[0] of instance: flextimer3. 101 Select mux mode: ALT5 mux port: CS1 of instance: dspi2. 111 Select mux mode: ALT7 mux port: debug_out[9] of instance: viu_mux.
19–14 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTD31 field descriptions (continued)

Field	Description
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD31. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD31. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD31. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD31. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD31. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD31. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD31. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD31. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD31. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD31.

Table continues on the next page...

IOMUXC_PTD31 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled

5.2.65 Software MUX Pad Control Register 64 (IOMUXC_PTD30)

Address: 4004_8000h base + 100h offset = 4004_8100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD30 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD30. 000 Select mux mode: ALT0 mux port: GPIO[64] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[30] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[14] of instance: nfc_mlc. 100 Select mux mode: ALT4 mux port: CH[1] of instance: flextimer3. 101 Select mux mode: ALT5 mux port: CS0 of instance: dspi2. 111 Select mux mode: ALT7 mux port: debug_out[10] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD30. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD30. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD30. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD30.

Table continues on the next page...

IOMUXC_PTD30 field descriptions (continued)

Field	Description
	0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD30. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD30. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD30. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD30. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD30. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD30. 0 Disabled 1 Enabled

5.2.66 Software MUX Pad Control Register 65 (IOMUXC_PTD29)

Address: 4004_8000h base + 104h offset = 4004_8104h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD29 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD29. 000 Select mux mode: ALT0 mux port: GPIO[65] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[29] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[13] of instance: nfc_mlc. 100 Select mux mode: ALT4 mux port: CH[2] of instance: flextimer3. 101 Select mux mode: ALT5 mux port: SIN of instance: dspi2. 111 Select mux mode: ALT7 mux port: debug_out[11] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD29. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD29. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD29. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD29. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD29. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD29. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD29.

Table continues on the next page...

IOMUXC_PTD29 field descriptions (continued)

Field	Description
	0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD29. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD29. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD29. 0 Disabled 1 Enabled

5.2.67 Software MUX Pad Control Register 66 (IOMUXC_PTD28)

Address: 4004_8000h base + 108h offset = 4004_8108h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD28 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTD28. **NOTE:** Pad PTD28 is involved in Daisy Chain. • Config Register IOMUXC_I2C2_IPP_SCL_IND_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[66] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[28] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[12] of instance: nfc_mlc. 011 Select mux mode: ALT3 mux port: SCL of instance: i2c2. 100 Select mux mode: ALT4 mux port: CH[3] of instance: flextimer3. 101 Select mux mode: ALT5 mux port: SOUT of instance: dspi2. 111 Select mux mode: ALT7 mux port: debug_out[12] of instance: viu_mux.
19–14 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTD28 field descriptions (continued)

Field	Description
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD28. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD28. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD28. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD28. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD28. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD28. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD28. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD28. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD28. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD28.

Table continues on the next page...

IOMUXC_PTD28 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled

5.2.68 Software MUX Pad Control Register 67 (IOMUXC_PTD27)

Address: 4004_8000h base + 10Ch offset = 4004_810Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD27 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTD27. **NOTE:** Pad PTD27 is involved in Daisy Chain. • Config Register IOMUXC_I2C2_IPP_SDA_IND_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[67] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[27] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[11] of instance: nfc_mlc. 011 Select mux mode: ALT3 mux port: SDA of instance: i2c2. 100 Select mux mode: ALT4 mux port: CH[4] of instance: flextimer3. 101 Select mux mode: ALT5 mux port: SCK of instance: dspi2. 111 Select mux mode: ALT7 mux port: debug_out[13] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD27. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD27. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD27.

Table continues on the next page...

IOMUXC_PTD27 field descriptions (continued)

Field	Description
	0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD27. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD27. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD27. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD27. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD27. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD27. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD27. 0 Disabled 1 Enabled

5.2.69 Software MUX Pad Control Register 68 (IOMUXC_PTD26)

Address: 4004_8000h base + 110h offset = 4004_8110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD26 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD26. 000 Select mux mode: ALT0 mux port: GPIO[68] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[26] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[10] of instance: nfc_mlc. 100 Select mux mode: ALT4 mux port: CH[5] of instance: flextimer3. 101 Select mux mode: ALT5 mux port: WP of instance: esdhc1. 111 Select mux mode: ALT7 mux port: debug_out[14] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD26. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD26. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD26. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD26. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD26. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTD26 field descriptions (continued)

Field	Description
	010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD26. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD26. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD26. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD26. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD26. 0 Disabled 1 Enabled

5.2.70 Software MUX Pad Control Register 69 (IOMUXC_PTD25)

Address: 4004_8000h base + 114h offset = 4004_8114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD25 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTD25 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTD25. 000 Select mux mode: ALT0 mux port: GPIO[69] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[25] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[9] of instance: nfc_mlc. 100 Select mux mode: ALT4 mux port: CH[6] of instance: flextimer3. 111 Select mux mode: ALT7 mux port: debug_out[15] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD25. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD25. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD25. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD25. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD25. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD25. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD25. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled

Table continues on the next page...

IOMUXC_PTD25 field descriptions (continued)

Field	Description
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD25. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD25. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD25. 0 Disabled 1 Enabled

5.2.71 Software MUX Pad Control Register 70 (IOMUXC_PTD24)

Address: 4004_8000h base + 118h offset = 4004_8118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD24 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTD24. 000 Select mux mode: ALT0 mux port: GPIO[70] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[24] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[8] of instance: nfc_mlc. 100 Select mux mode: ALT4 mux port: CH[7] of instance: flextimer3. 111 Select mux mode: ALT7 mux port: debug_out[16] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD24. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)

Table continues on the next page...

IOMUXC_PTD24 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD24. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD24. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD24. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD24. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD24. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD24. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD24. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD24. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD24. 0 Disabled 1 Enabled

5.2.72 Software MUX Pad Control Register 71 (IOMUXC_PTD23)

Address: 4004_8000h base + 11Ch offset = 4004_811Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD23 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTD23. **NOTE:** Pad PTD23 is involved in Daisy Chain. • Config Register IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_0_SELECT_INPUT for mode ALT4. • Config Register IOMUXC_SCI_FLX2_IPP_IND_SCI_TX_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: GPIO[71] of instance: rgpioc. Also, RXDATA[3] for MAC0 is enabled in this mux mode so ensure obe is disabled if this pin is used for MAC0-MII instead of GPIO. 001 Select mux mode: ALT1 mux port: FB_AD[23] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[7] of instance: nfc_mlc. 011 Select mux mode: ALT3 mux port: CH[0] of instance: flextimer2. 100 Select mux mode: ALT4 mux port: MAC0_TMR0 of instance: enet_swiahb. 101 Select mux mode: ALT5 mux port: DAT4 of instance: esdhc0. 110 Select mux mode: ALT6 mux port: TX of instance: sci_flx2. 111 Select mux mode: ALT7 mux port: DATA_OUT[21] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD23. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD23. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD23. 0 Output is CMOS 1 Output is open drain

Table continues on the next page...

IOMUXC_PTD23 field descriptions (continued)

Field	Description
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD23. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD23. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD23. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD23. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD23. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD23. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD23. 0 Disabled 1 Enabled

5.2.73 Software MUX Pad Control Register 72 (IOMUXC_PTD22)

Address: 4004_8000h base + 120h offset = 4004_8120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD22 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTD22. **NOTE:** Pad PTD22 is involved in Daisy Chain. • Config Register IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_1_SELECT_INPUT for mode ALT4. • Config Register IOMUXC_SCI_FLX2_IPP_IND_SCI_RX_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: GPIO[72] of instance: rgpioc. Also, RXDATA[2] for MAC0 is enabled in this mux mode so ensure obe is disabled if this pin is used for MAC0-MII instead of GPIO. 001 Select mux mode: ALT1 mux port: FB_AD[22] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[6] of instance: nfc_mlc. 011 Select mux mode: ALT3 mux port: CH[1] of instance: flextimer2. 100 Select mux mode: ALT4 mux port: MAC0_TMR1 of instance: enet_swiahb. 101 Select mux mode: ALT5 mux port: DAT5 of instance: esdhc0. 110 Select mux mode: ALT6 mux port: RX of instance: sci_flx2. 111 Select mux mode: ALT7 mux port: DATA_OUT[22] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD22. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD22. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD22. 0 Output is CMOS 1 Output is open drain

Table continues on the next page...

IOMUXC_PTD22 field descriptions (continued)

Field	Description
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD22. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD22. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD22. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD22. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD22. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD22. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD22. 0 Disabled 1 Enabled

5.2.74 Software MUX Pad Control Register 73 (IOMUXC_PTD21)

Address: 4004_8000h base + 124h offset = 4004_8124h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD21 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTD21. 000 Select mux mode: ALT0 mux port: GPIO[73] of instance: rgpioc. Also, CRS for MAC0 is enabled in this mux mode so ensure obe is disabled if this pin is used for MAC0-MII instead of GPIO. 001 Select mux mode: ALT1 mux port: FB_AD[21] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[5] of instance: nfc_mlc. 100 Select mux mode: ALT4 mux port: MAC0_TMR2 of instance: enet_swiahb. 101 Select mux mode: ALT5 mux port: DAT6 of instance: esdhc0. 110 Select mux mode: ALT6 mux port: RTS of instance: sci_flx2. 111 Select mux mode: ALT7 mux port: DATA_OUT[23] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD21. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD21. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD21. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD21. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD21.

Table continues on the next page...

IOMUXC_PTD21 field descriptions (continued)

Field	Description
	000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD21. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD21. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD21. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD21. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD21. 0 Disabled 1 Enabled

5.2.75 Software MUX Pad Control Register 74 (IOMUXC_PTD20)

Address: 4004_8000h base + 128h offset = 4004_8128h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD20 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTD20. **NOTE:** Pad PTD20 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX2_IPP_IND_CTS_B_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: GPIO[74] of instance: rgpioc. Also, COL for MAC0 is enabled in this mux mode so ensure obe is disabled if this pin is used for MAC0-MII instead of GPIO. 001 Select mux mode: ALT1 mux port: FB_AD[20] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[4] of instance: nfc_mlc. 100 Select mux mode: ALT4 mux port: MAC0_TMR3 of instance: enet_swiahb. 101 Select mux mode: ALT5 mux port: DAT7 of instance: esdhc0. 110 Select mux mode: ALT6 mux port: CTS of instance: sci_flx2. 111 Select mux mode: ALT7 mux port: DATA_OUT[18] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD20. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD20. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD20. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD20. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD20. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD20. 00 100 kOhm Pull Down 01 47 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTD20 field descriptions (continued)

Field	Description
	10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD20. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD20. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD20. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD20. 0 Disabled 1 Enabled

5.2.76 Software MUX Pad Control Register 75 (IOMUXC_PTD19)

Address: 4004_8000h base + 12Ch offset = 4004_812Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD19 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTD19. **NOTE:** Pad PTD19 is involved in Daisy Chain. • Config Register IOMUXC_I2C0_IPP_SCL_IND_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: GPIO[75] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[19] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[3] of instance: nfc_mlc. 011 Select mux mode: ALT3 mux port: SCKR of instance: esai. 100 Select mux mode: ALT4 mux port: SCL of instance: i2c0.

Table continues on the next page...

IOMUXC_PTD19 field descriptions (continued)

Field	Description
	101 Select mux mode: ALT5 mux port: QD_PHA of instance: flextimer2. 110 Select mux mode: ALT6 mux port: TXDATA[3] for MAC0-MII 111 Select mux mode: ALT7 mux port: DATA_OUT[19] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD19. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD19. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD19. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD19. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD19. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD19. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD19. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD19. 0 Keeper enable 1 Pull enable

Table continues on the next page...

IOMUXC_PTD19 field descriptions (continued)

Field	Description
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD19. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD19. 0 Disabled 1 Enabled

5.2.77 Software MUX Pad Control Register 76 (IOMUXC_PTD18)

Address: 4004_8000h base + 130h offset = 4004_8130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD18 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTD18. **NOTE:** Pad PTD18 is involved in Daisy Chain. • Config Register IOMUXC_I2C0_IPP_SDA_IND_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: GPIO[76] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[18] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[2] of instance: nfc_mlc. 011 Select mux mode: ALT3 mux port: FSR of instance: esai. 100 Select mux mode: ALT4 mux port: SDA of instance: i2c0. 101 Select mux mode: ALT5 mux port: QD_PHB of instance: flextimer2. 111 Select mux mode: ALT7 mux port: DATA_OUT[10] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD18. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)

Table continues on the next page...

IOMUXC_PTD18 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD18. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD18. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD18. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD18. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD18. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD18. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD18. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD18. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD18. 0 Disabled 1 Enabled

5.2.78 Software MUX Pad Control Register 77 (IOMUXC_PTD17)

Address: 4004_8000h base + 134h offset = 4004_8134h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD17 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD17. **NOTE:** Pad PTD17 is involved in Daisy Chain. • Config Register IOMUXC_I2C1_IPP_SCL_IND_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: GPIO[77] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[17] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[1] of instance: nfc_mlc. 011 Select mux mode: ALT3 mux port: HCKR of instance: esai. 100 Select mux mode: ALT4 mux port: SCL of instance: i2c1. 110 Select mux mode: ALT6 mux port: TXERR of MAC0-MII 111 Select mux mode: ALT7 mux port: DATA_OUT[11] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD17. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD17. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD17. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD17. 0 CMOS input 1 Schmitt trigger input

Table continues on the next page...

IOMUXC_PTD17 field descriptions (continued)

Field	Description
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD17. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD17. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD17. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD17. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD17. 0 Disabled 1 Enabled

5.2.79 Software MUX Pad Control Register 78 (IOMUXC_PTD16)

Address: 4004_8000h base + 138h offset = 4004_8138h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD16 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD16. **NOTE:** Pad PTD16 is involved in Daisy Chain. • Config Register IOMUXC_I2C1_IPP_SDA_IND_SELECT_INPUT for mode ALT4. 000 Select mux mode: ALT0 mux port: GPIO[78] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: FB_AD[16] of instance: platform. 010 Select mux mode: ALT2 mux port: NF_IO[0] of instance: nfc_mlc. 011 Select mux mode: ALT3 mux port: HCKT of instance: esai. 100 Select mux mode: ALT4 mux port: SDA of instance: i2c1. 111 Select mux mode: ALT7 mux port: DATA_OUT[12] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD16. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD16. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD16. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD16. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD16. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD16. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTD16 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD16. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD16. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD16. 0 Disabled 1 Enabled

5.2.80 Software MUX Pad Control Register 79 (IOMUXC_PTD0)

Address: 4004_8000h base + 13Ch offset = 4004_813Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD0 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD0. **NOTE:** Pad PTD0 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX2_IPP_IND_SCI_TX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[79] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSCK_A of instance: quadspi0. 010 Select mux mode: ALT2 mux port: TX of instance: sci_flx2. 100 Select mux mode: ALT4 mux port: FB_AD[15] of instance: platform. 101 Select mux mode: ALT5 mux port: EXTCLK of instance: spdif. 111 Select mux mode: ALT7 mux port: debug_out[17] of instance: viu_mux.
19–14 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTD0 field descriptions (continued)

Field	Description
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD0. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD0. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD0. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD0. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD0. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD0.

Table continues on the next page...

IOMUXC_PTD0 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled

5.2.81 Software MUX Pad Control Register 80 (IOMUXC_PTD1)

Address: 4004_8000h base + 140h offset = 4004_8140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD1 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD1. **NOTE:** Pad PTD1 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX2_IPP_IND_SCI_RX_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[80] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QPCS0_A of instance: quadspi0. 010 Select mux mode: ALT2 mux port: RX of instance: sci_flx2. 100 Select mux mode: ALT4 mux port: FB_AD[14] of instance: platform. 101 Select mux mode: ALT5 mux port: IN1 of instance: spdif. 111 Select mux mode: ALT7 mux port: debug_out[18] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD1. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD1. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD1.

Table continues on the next page...

IOMUXC_PTD1 field descriptions (continued)

Field	Description
	0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD1. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD1. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD1. 0 Disabled 1 Enabled

5.2.82 Software MUX Pad Control Register 81 (IOMUXC_PTD2)

Address: 4004_8000h base + 144h offset = 4004_8144h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD2 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTD2. 000 Select mux mode: ALT0 mux port: GPIO[81] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSPI_IO3_A of instance: quadspi0. 010 Select mux mode: ALT2 mux port: RTS of instance: sci_flx2. 011 Select mux mode: ALT3 mux port: CS3 of instance: dspi1. 100 Select mux mode: ALT4 mux port: FB_AD[13] of instance: platform. 101 Select mux mode: ALT5 mux port: OUT1 of instance: spdif. 111 Select mux mode: ALT7 mux port: debug_out[19] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD2. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD2. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD2. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD2. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD2. 000 output driver disabled;

Table continues on the next page...

IOMUXC_PTD2 field descriptions (continued)

Field	Description
	001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD2. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD2. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD2. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD2. 0 Disabled 1 Enabled

5.2.83 Software MUX Pad Control Register 82 (IOMUXC_PTD3)

Address: 4004_8000h base + 148h offset = 4004_8148h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD3 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTD3. **NOTE:** Pad PTD3 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX2_IPP_IND_CTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[82] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSPI_IO2_A of instance: quadspi0. 010 Select mux mode: ALT2 mux port: CTS of instance: sci_flx2. 011 Select mux mode: ALT3 mux port: CS2 of instance: dspi1. 100 Select mux mode: ALT4 mux port: FB_AD[12] of instance: platform. 101 Select mux mode: ALT5 mux port: PLOCK of instance: spdif. 111 Select mux mode: ALT7 mux port: debug_out[20] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD3. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD3. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD3. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD3. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD3. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD3. 00 100 kOhm Pull Down 01 47 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTD3 field descriptions (continued)

Field	Description
	10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD3. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD3. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD3. 0 Disabled 1 Enabled

5.2.84 Software MUX Pad Control Register 83 (IOMUXC_PTD4)

Address: 4004_8000h base + 14Ch offset = 4004_814Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD4 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD4. 000 Select mux mode: ALT0 mux port: GPIO[83] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSPI_IO1_A of instance: quadspi0. 011 Select mux mode: ALT3 mux port: CS1 of instance: dspi1. 100 Select mux mode: ALT4 mux port: FB_AD[11] of instance: platform. 101 Select mux mode: ALT5 mux port: SRCLK of instance: spdif. 111 Select mux mode: ALT7 mux port: debug_out[21] of instance: viu_mux.
19–14 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTD4 field descriptions (continued)

Field	Description
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD4. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD4. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD4. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD4. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD4. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD4. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD4. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD4. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD4.

Table continues on the next page...

IOMUXC_PTD4 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled

5.2.85 Software MUX Pad Control Register 84 (IOMUXC_PTD5)

Address: 4004_8000h base + 150h offset = 4004_8150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD5 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTD5. **NOTE:** Pad PTD5 is involved in Daisy Chain. • Config Register IOMUXC_DSPI1_IPP_IND_SS_B_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[84] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSPI_IO0_A of instance: quadspi0. 011 Select mux mode: ALT3 mux port: CS0 of instance: dspi1. 100 Select mux mode: ALT4 mux port: FB_AD[10] of instance: platform. 111 Select mux mode: ALT7 mux port: debug_out[22] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD5. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD5. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD5. 0 Output is CMOS 1 Output is open drain

Table continues on the next page...

IOMUXC_PTD5 field descriptions (continued)

Field	Description
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD5. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD5. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD5. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD5. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD5. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD5. 0 Disabled 1 Enabled

5.2.86 Software MUX Pad Control Register 85 (IOMUXC_PTD6)

Address: 4004_8000h base + 154h offset = 4004_8154h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD6 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTD6. **NOTE:** Pad PTD6 is involved in Daisy Chain. • Config Register IOMUXC_DSPI1_IPP_IND_SIN_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[85] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DQS_A of instance: quadspi0. 011 Select mux mode: ALT3 mux port: SIN of instance: dspi1. 100 Select mux mode: ALT4 mux port: FB_AD[9] of instance: platform. 111 Select mux mode: ALT7 mux port: debug_out[23] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD6. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD6. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD6. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD6. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD6. 000 output driver disabled;

Table continues on the next page...

IOMUXC_PTD6 field descriptions (continued)

Field	Description
	001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD6. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD6. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD6. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD6. 0 Disabled 1 Enabled

5.2.87 Software MUX Pad Control Register 86 (IOMUXC_PTD7)

Address: 4004_8000h base + 158h offset = 4004_8158h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD7 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTD7. 000 Select mux mode: ALT0 mux port: GPIO[86] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSCK_B of instance: quadspi0. 011 Select mux mode: ALT3 mux port: SOUT of instance: dspi1. 100 Select mux mode: ALT4 mux port: FB_AD[8] of instance: platform. 111 Select mux mode: ALT7 mux port: debug_out[24] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD7. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD7. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD7. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD7. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD7. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD7. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD7.

Table continues on the next page...

IOMUXC_PTD7 field descriptions (continued)

Field	Description
	0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD7. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD7. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD7. 0 Disabled 1 Enabled

5.2.88 Software MUX Pad Control Register 87 (IOMUXC_PTD8)

Address: 4004_8000h base + 15Ch offset = 4004_815Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD8 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD8. **NOTE:** Pad PTD8 is involved in Daisy Chain. • Config Register IOMUXC_DSPI1_IPP_IND_SCK_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[87] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QPCS0_B of instance: quadspi0. 010 Select mux mode: ALT2 mux port: FB_CLKOUT of instance: lpcg0. 011 Select mux mode: ALT3 mux port: SCK of instance: dspi1. 100 Select mux mode: ALT4 mux port: FB_AD[7] of instance: platform. 111 Select mux mode: ALT7 mux port: debug_out[25] of instance: viu_mux.
19–14 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTD8 field descriptions (continued)

Field	Description
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD8. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD8. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD8. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD8. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD8. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD8. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD8. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD8. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD8.

Table continues on the next page...

IOMUXC_PTD8 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled

5.2.89 Software MUX Pad Control Register 88 (IOMUXC_PTD9)

Address: 4004_8000h base + 160h offset = 4004_8160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD9 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTD9. **NOTE:** Pad PTD9 is involved in Daisy Chain. • Config Register IOMUXC_SAI1_IPP_IND_SAI_TXSYNC_SELECT_INPUT for mode ALT6. 000 Select mux mode: ALT0 mux port: GPIO[88] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSPI_IO3_B of instance: quadspi0. 010 Select mux mode: ALT2 mux port: CS1 of instance: dspi3. 100 Select mux mode: ALT4 mux port: FB_AD[6] of instance: platform. 110 Select mux mode: ALT6 mux port: TX_SYNC of instance: sai1. 111 Select mux mode: ALT7 mux port: DATA_OUT[2] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD9. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD9. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD9.

Table continues on the next page...

IOMUXC_PTD9 field descriptions (continued)

Field	Description
	0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD9. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD9. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD9. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD9. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD9. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD9. 0 Disabled 1 Enabled

5.2.90 Software MUX Pad Control Register 89 (IOMUXC_PTD10)

Address: 4004_8000h base + 164h offset = 4004_8164h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD10 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTD10. 000 Select mux mode: ALT0 mux port: GPIO[89] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSPI_IO2_B of instance: quadspi0. 010 Select mux mode: ALT2 mux port: CS0 of instance: dspi3. 100 Select mux mode: ALT4 mux port: FB_AD[5] of instance: platform. 111 Select mux mode: ALT7 mux port: DATA_OUT[3] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD10. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD10. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD10. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD10. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD10. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTD10 field descriptions (continued)

Field	Description
	011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD10. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD10. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD10. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD10. 0 Disabled 1 Enabled

5.2.91 Software MUX Pad Control Register 90 (IOMUXC_PTD11)

Address: 4004_8000h base + 168h offset = 4004_8168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD11 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTD11 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTD11. 000 Select mux mode: ALT0 mux port: GPIO[90] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSPI_IO1_B of instance: quadspi0. 010 Select mux mode: ALT2 mux port: SIN of instance: dspi3. 100 Select mux mode: ALT4 mux port: FB_AD[4] of instance: platform. 111 Select mux mode: ALT7 mux port: debug_out[26] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD11. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD11. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD11. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD11. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD11. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD11. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled

Table continues on the next page...

IOMUXC_PTD11 field descriptions (continued)

Field	Description
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD11. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD11. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD11. 0 Disabled 1 Enabled

5.2.92 Software MUX Pad Control Register 91 (IOMUXC_PTD12)

Address: 4004_8000h base + 16Ch offset = 4004_816Ch

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>R
W</td><td colspan="9">Reserved</td><td colspan="3">MUX_MODE</td><td colspan="4">Reserved</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R
W</td><td colspan="2">Reserved</td><td colspan="2">SPEED</td><td>SRE</td><td>ODE</td><td>HYS</td><td colspan="3">DSE</td><td colspan="2">PUS</td><td>PKE</td><td>PUE</td><td>OBE</td><td>IBE</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

IOMUXC_PTD12 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTD12. 000 Select mux mode: ALT0 mux port: GPIO[91] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: QSPI_IO0_B of instance: quadspi0. 010 Select mux mode: ALT2 mux port: SOUT of instance: dspi3. 100 Select mux mode: ALT4 mux port: FB_AD[3] of instance: platform. 111 Select mux mode: ALT7 mux port: debug_out[27] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD12. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)

Table continues on the next page...

IOMUXC_PTD12 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD12. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD12. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD12. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD12. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD12. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD12. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD12. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD12. 0 Disabled 1 Enabled

5.2.93 Software MUX Pad Control Register 92 (IOMUXC_PTD13)

Address: 4004_8000h base + 170h offset = 4004_8170h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTD13 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTD13. 000 Select mux mode: ALT0 mux port: GPIO[92] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DQS_B of instance: quadspi0. 010 Select mux mode: ALT2 mux port: SCK of instance: dspi3. 100 Select mux mode: ALT4 mux port: FB_AD[2] of instance: platform. 111 Select mux mode: ALT7 mux port: debug_out[28] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTD13. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTD13. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTD13. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTD13. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTD13. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTD13 field descriptions (continued)

Field	Description
	011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTD13. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTD13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTD13. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTD13. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTD13. 0 Disabled 1 Enabled

5.2.94 Software MUX Pad Control Register 93 (IOMUXC_PTB23)

Address: 4004_8000h base + 174h offset = 4004_8174h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTB23 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTB23 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTB23. **NOTE:** Pad PTB23 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX1_IPP_IND_SCI_TX_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_SRC_IPP_BOOT_CFG_18_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[93] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TX_BCLK of instance: sai0. 010 Select mux mode: ALT2 mux port: TX of instance: sci_flx1. 100 Select mux mode: ALT4 mux port: FB_MUXED_ALE of instance: platform. 101 Select mux mode: ALT5 mux port: FB_TS_b of instance: platform. 110 Select mux mode: ALT6 mux port: RTS of instance: sci_flx3. 111 Select mux mode: ALT7 mux port: DATA_OUT[13] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB23. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB23. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB23. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB23. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB23. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB23. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTB23 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB23. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB23. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB23. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB23. 0 Disabled 1 Enabled

5.2.95 Software MUX Pad Control Register 94 (IOMUXC_PTB24)

Address: 4004_8000h base + 178h offset = 4004_8178h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTB24 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTB24. **NOTE:** Pad PTB24 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX1_IPP_IND_SCI_RX_SELECT_INPUT for mode ALT2. • Config Register IOMUXC_SRC_IPP_BOOT_CFG_19_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[94] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RX_BCLK of instance: sai0. 010 Select mux mode: ALT2 mux port: RX of instance: sci_flx1. 100 Select mux mode: ALT4 mux port: FB_MUXED_TSIZ0 of instance: platform. 101 Select mux mode: ALT5 mux port: NF_WE_b of instance: nfc_mlc. 110 Select mux mode: ALT6 mux port: CTS of instance: sci_flx3. 111 Select mux mode: ALT7 mux port: DATA_OUT[14] of instance: tcon1.

Table continues on the next page...

IOMUXC_PTB24 field descriptions (continued)

Field	Description
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB24. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB24. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB24. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB24. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB24. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB24. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB24. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB24. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB24. 0 Disabled 1 Enabled

Table continues on the next page...

IOMUXC_PTB24 field descriptions (continued)

Field	Description
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB24. 0 Disabled 1 Enabled

5.2.96 Software MUX Pad Control Register 95 (IOMUXC_PTB25)

Address: 4004_8000h base + 17Ch offset = 4004_817Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTB25 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTB25. **NOTE:** Pad PTB25 is involved in Daisy Chain. • Config Register IOMUXC_SRC_IPP_BOOT_CFG_20_SELECT_INPUT for mode ALT3. 000 Select mux mode: ALT0 mux port: GPIO[95] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RX_DATA of instance: sai0. 010 Select mux mode: ALT2 mux port: RTS of instance: sci_flx1. 100 Select mux mode: ALT4 mux port: FB_CS1_b of instance: platform. 101 Select mux mode: ALT5 mux port: NF_CE0_b of instance: nfc_mlc. 111 Select mux mode: ALT7 mux port: DATA_OUT[15] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB25. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB25. 0 Slow Slew Rate 1 Fast Slew Rate

Table continues on the next page...

IOMUXC_PTB25 field descriptions (continued)

Field	Description
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB25. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB25. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB25. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB25. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB25. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB25. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB25. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB25. 0 Disabled 1 Enabled

5.2.97 Software MUX Pad Control Register 96 (IOMUXC_PTB26)

Address: 4004_8000h base + 180h offset = 4004_8180h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTB26 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTB26. **NOTE:** Pad PTB26 is involved in Daisy Chain. • Config Register IOMUXC_SCI_FLX1_IPP_IND_CTS_B_SELECT_INPUT for mode ALT2. 000 Select mux mode: ALT0 mux port: GPIO[96] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TX_DATA of instance: sai0. 010 Select mux mode: ALT2 mux port: CTS of instance: sci_flx1. 011 Select mux mode: ALT3 mux port: RCON21 of instance: src. 100 Select mux mode: ALT4 mux port: FB_CS0_b of instance: platform. 101 Select mux mode: ALT5 mux port: NF_CE1_b of instance: nfc_mlc. 111 Select mux mode: ALT7 mux port: DATA_OUT[16] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB26. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB26. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB26. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB26. 0 CMOS input 1 Schmitt trigger input

Table continues on the next page...

IOMUXC_PTB26 field descriptions (continued)

Field	Description
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB26. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB26. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB26. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB26. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB26. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB26. 0 Disabled 1 Enabled

5.2.98 Software MUX Pad Control Register 97 (IOMUXC_PTB27)

Address: 4004_8000h base + 184h offset = 4004_8184h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTB27 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTB27. 000 Select mux mode: ALT0 mux port: GPIO[97] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RX_SYNC of instance: sai0. 011 Select mux mode: ALT3 mux port: RCON22 of instance: src. 100 Select mux mode: ALT4 mux port: FB_OE_b of instance: platform. 101 Select mux mode: ALT5 mux port: FB_MUXED_TBST_b of instance: platform. 110 Select mux mode: ALT6 mux port: NF_RE_b of instance: nfc_mlc. 111 Select mux mode: ALT7 mux port: DATA_OUT[17] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB27. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB27. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB27. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB27. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB27. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB27. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTB27 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB27. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB27. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB27. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB27. 0 Disabled 1 Enabled

5.2.99 Software MUX Pad Control Register 98 (IOMUXC_PTB28)

Address: 4004_8000h base + 188h offset = 4004_8188h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTB28 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTB28. 000 Select mux mode: ALT0 mux port: GPIO[98] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TX_SYNC of instance: sai0. 011 Select mux mode: ALT3 mux port: RCON23 of instance: src. 100 Select mux mode: ALT4 mux port: FB_RW_b of instance: platform. 111 Select mux mode: ALT7 mux port: DATA_OUT[8] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTB28. 00 Low (50 MHz)

Table continues on the next page...

IOMUXC_PTB28 field descriptions (continued)

Field	Description
	01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTB28. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTB28. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTB28. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTB28. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTB28. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTB28. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTB28. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTB28. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTB28. 0 Disabled 1 Enabled

5.2.100 Software MUX Pad Control Register 99 (IOMUXC_PTC26)

Address: 4004_8000h base + 18Ch offset = 4004_818Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTC26 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTC26. 000 Select mux mode: ALT0 mux port: GPIO[99] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TX_BCLK of instance: sai1. 010 Select mux mode: ALT2 mux port: CS5 of instance: dspi0. 011 Select mux mode: ALT3 mux port: RCON24 of instance: src. 100 Select mux mode: ALT4 mux port: FB_TA_b of instance: platform. 101 Select mux mode: ALT5 mux port: NF_RB_b of instance: nfc_mlc. 111 Select mux mode: ALT7 mux port: DATA_OUT[9] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC26. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC26. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC26. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC26. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC26. 000 output driver disabled;

Table continues on the next page...

IOMUXC_PTC26 field descriptions (continued)

Field	Description
	001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC26. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC26. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC26. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC26. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC26. 0 Disabled 1 Enabled

5.2.101 Software MUX Pad Control Register 100 (IOMUXC_PTC27)

Address: 4004_8000h base + 190h offset = 4004_8190h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTC27 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTC27. 000 Select mux mode: ALT0 mux port: GPIO[100] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RX_BCLK of instance: sai1. 010 Select mux mode: ALT2 mux port: CS4 of instance: dspi0. 011 Select mux mode: ALT3 mux port: RCON25 of instance: src. 100 Select mux mode: ALT4 mux port: FB_BE3_b of instance: platform. 101 Select mux mode: ALT5 mux port: FB_CS3_b of instance: platform. 110 Select mux mode: ALT6 mux port: NF_ALE of instance: nfc_mlc. 111 Select mux mode: ALT7 mux port: DATA_OUT[4] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC27. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC27. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC27. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC27. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC27. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC27. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTC27 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC27. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC27. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC27. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC27. 0 Disabled 1 Enabled

5.2.102 Software MUX Pad Control Register 101 (IOMUXC_PTC28)

Address: 4004_8000h base + 194h offset = 4004_8194h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTC28 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTC28. 000 Select mux mode: ALT0 mux port: GPIO[101] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RX_DATA of instance: sai1. 010 Select mux mode: ALT2 mux port: CS3 of instance: dspi0. 011 Select mux mode: ALT3 mux port: RCON26 of instance: src. 100 Select mux mode: ALT4 mux port: FB_BE2_b of instance: platform. 101 Select mux mode: ALT5 mux port: FB_CS2_b of instance: platform. 110 Select mux mode: ALT6 mux port: NF_CLE of instance: nfc_mlc. 111 Select mux mode: ALT7 mux port: DATA_OUT[5] of instance: tcon1.
19–14 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTC28 field descriptions (continued)

Field	Description
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC28. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC28. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC28. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC28. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC28. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC28. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC28. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC28. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC28. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC28.

Table continues on the next page...

IOMUXC_PTC28 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled

5.2.103 Software MUX Pad Control Register 102 (IOMUXC_PTC29)

Address: 4004_8000h base + 198h offset = 4004_8198h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTC29 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 7 IOMUX modes to be used for pad: PTC29. 000 Select mux mode: ALT0 mux port: GPIO[102] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TX_DATA of instance: sai1. 010 Select mux mode: ALT2 mux port: CS2 of instance: dspi0. 011 Select mux mode: ALT3 mux port: RCON27 of instance: src. 100 Select mux mode: ALT4 mux port: FB_BE1_b of instance: platform. 101 Select mux mode: ALT5 mux port: FB_MUXED_TSIZ1 of instance: platform. 111 Select mux mode: ALT7 mux port: DATA_OUT[6] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC29. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC29. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC29. 0 Output is CMOS 1 Output is open drain

Table continues on the next page...

IOMUXC_PTC29 field descriptions (continued)

Field	Description
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC29. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC29. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC29. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC29. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC29. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC29. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC29. 0 Disabled 1 Enabled

5.2.104 Software MUX Pad Control Register 103 (IOMUXC_PTC30)

Address: 4004_8000h base + 19Ch offset = 4004_819Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTC30 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 8 IOMUX modes to be used for pad: PTC30. 000 Select mux mode: ALT0 mux port: GPIO[103] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: RX_SYNC of instance: sai1. 010 Select mux mode: ALT2 mux port: CS2 of instance: dspi1. 011 Select mux mode: ALT3 mux port: RCON28 of instance: src. 100 Select mux mode: ALT4 mux port: FB_MUXED_BE0_b of instance: platform. 101 Select mux mode: ALT5 mux port: FB_TSIZ0 of instance: platform. 110 Select mux mode: ALT6 mux port: ADC0SE5 of instance: adc0_da. 111 Select mux mode: ALT7 mux port: DATA_OUT[7] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC30. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC30. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC30. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC30. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC30.

Table continues on the next page...

IOMUXC_PTC30 field descriptions (continued)

Field	Description
	000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC30. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC30. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC30. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC30. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC30. 0 Disabled 1 Enabled

5.2.105 Software MUX Pad Control Register 104 (IOMUXC_PTC31)

Address: 4004_8000h base + 1A0h offset = 4004_81A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

IOMUXC_PTC31 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTC31. **NOTE:** Pad PTC31 is involved in Daisy Chain. • Config Register IOMUXC_SAI1_IPP_IND_SAI_TXSYNC_SELECT_INPUT for mode ALT1. 000 Select mux mode: ALT0 mux port: GPIO[104] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TX_SYNC of instance: sai1. 011 Select mux mode: ALT3 mux port: RCON29 of instance: src. 110 Select mux mode: ALT6 mux port: ADC1SE5 of instance: adc1_da. 111 Select mux mode: ALT7 mux port: DATA_OUT[8] of instance: tcon1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTC31. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTC31. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTC31. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTC31. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTC31. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTC31. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTC31 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTC31. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTC31. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTC31. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTC31. 0 Disabled 1 Enabled

5.2.106 Software MUX Pad Control Register 105 (IOMUXC_PTE0)

Address: 4004_8000h base + 1A4h offset = 4004_81A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0

IOMUXC_PTE0 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE0. 000 Select mux mode: ALT0 mux port: GPIO[105] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TCON[1] of instance: tcon0. 010 Select mux mode: ALT2 mux port: BOOTMODE[1] of instance: src. 100 Select mux mode: ALT4 mux port: LCD0 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[29] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE0. 00 Low (50 MHz)

Table continues on the next page...

IOMUXC_PTE0 field descriptions (continued)

Field	Description
	01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE0. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE0. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE0. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE0. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE0. 0 Disabled 1 Enabled

5.2.107 Software MUX Pad Control Register 106 (IOMUXC_PTE1)

Address: 4004_8000h base + 1A8h offset = 4004_81A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0

IOMUXC_PTE1 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE1. 000 Select mux mode: ALT0 mux port: GPIO[106] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TCON[2] of instance: tcon0. 010 Select mux mode: ALT2 mux port: BOOTMODE[0] of instance: src. 100 Select mux mode: ALT4 mux port: LCD1 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[30] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE1. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE1. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE1. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTE1 field descriptions (continued)

Field	Description
	011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE1. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE1. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE1. 0 Disabled 1 Enabled

5.2.108 Software MUX Pad Control Register 107 (IOMUXC_PTE2)

Address: 4004_8000h base + 1ACh offset = 4004_81ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE2 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTE2 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE2. 000 Select mux mode: ALT0 mux port: GPIO[107] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[1] of instance: tcon0. 100 Select mux mode: ALT4 mux port: LCD2 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[31] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE2. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE2. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE2. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE2. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE2. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE2. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE2.

Table continues on the next page...

IOMUXC_PTE2 field descriptions (continued)

Field	Description
	0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE2. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE2. 0 Disabled 1 Enabled

5.2.109 Software MUX Pad Control Register 108 (IOMUXC_PTE3)

Address: 4004_8000h base + 1B0h offset = 4004_81B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE3 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE3. 000 Select mux mode: ALT0 mux port: GPIO[108] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TCON[0] of instance: tcon0. 100 Select mux mode: ALT4 mux port: LCD3 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[32] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE3. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE3. 0 Slow Slew Rate 1 Fast Slew Rate

Table continues on the next page...

IOMUXC_PTE3 field descriptions (continued)

Field	Description
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE3. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE3. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE3. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE3. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE3. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE3. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE3. 0 Disabled 1 Enabled

5.2.110 Software MUX Pad Control Register 109 (IOMUXC_PTE4)

Address: 4004_8000h base + 1B4h offset = 4004_81B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE4 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE4. 000 Select mux mode: ALT0 mux port: GPIO[109] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: TCON[3] of instance: tcon0. 100 Select mux mode: ALT4 mux port: LCD4 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[33] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE4. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE4. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE4. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE4. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE4. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTE4 field descriptions (continued)

Field	Description
	100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE4. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE4. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE4. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE4. 0 Disabled 1 Enabled

5.2.111 Software MUX Pad Control Register 110 (IOMUXC_PTE5)

Address: 4004_8000h base + 1B8h offset = 4004_81B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE5 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE5.

Table continues on the next page...

IOMUXC_PTE5 field descriptions (continued)

Field	Description
	000 Select mux mode: ALT0 mux port: GPIO[110] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[18] of instance: tcon0. 100 Select mux mode: ALT4 mux port: LCD5 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[34] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE5. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE5. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE5. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE5. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE5. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE5. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE5. 0 Keeper enable 1 Pull enable

Table continues on the next page...

IOMUXC_PTE5 field descriptions (continued)

Field	Description
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE5. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE5. 0 Disabled 1 Enabled

5.2.112 Software MUX Pad Control Register 111 (IOMUXC_PTE6)

Address: 4004_8000h base + 1BCh offset = 4004_81BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE6 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE6. 000 Select mux mode: ALT0 mux port: GPIO[111] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[19] of instance: tcon0. 100 Select mux mode: ALT4 mux port: LCD6 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[35] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE6. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE6. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE6.

Table continues on the next page...

IOMUXC_PTE6 field descriptions (continued)

Field	Description
	0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE6. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE6. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE6. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE6. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE6. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE6. 0 Disabled 1 Enabled

5.2.113 Software MUX Pad Control Register 112 (IOMUXC_PTE7)

Address: 4004_8000h base + 1C0h offset = 4004_81C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE7 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE7. 000 Select mux mode: ALT0 mux port: GPIO[112] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[20] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON0 of instance: src. 100 Select mux mode: ALT4 mux port: LCD7 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[36] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE7. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE7. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE7. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE7. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE7. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTE7 field descriptions (continued)

Field	Description
	011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE7. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE7. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE7. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE7. 0 Disabled 1 Enabled

5.2.114 Software MUX Pad Control Register 113 (IOMUXC_PTE8)

Address: 4004_8000h base + 1C4h offset = 4004_81C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE8 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTE8 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE8. 000 Select mux mode: ALT0 mux port: GPIO[113] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[21] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON1 of instance: src. 100 Select mux mode: ALT4 mux port: LCD8 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[37] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE8. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE8. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE8. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE8. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE8. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE8. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled

Table continues on the next page...

IOMUXC_PTE8 field descriptions (continued)

Field	Description
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE8. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE8. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE8. 0 Disabled 1 Enabled

5.2.115 Software MUX Pad Control Register 114 (IOMUXC_PTE9)

Address: 4004_8000h base + 1C8h offset = 4004_81C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE9 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE9. 000 Select mux mode: ALT0 mux port: GPIO[114] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[22] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON2 of instance: src. 100 Select mux mode: ALT4 mux port: LCD9 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[38] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE9. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)

Table continues on the next page...

IOMUXC_PTE9 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE9. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE9. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE9. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE9. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE9. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE9. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE9. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE9. 0 Disabled 1 Enabled

5.2.116 Software MUX Pad Control Register 115 (IOMUXC_PTE10)

Address: 4004_8000h base + 1CCh offset = 4004_81CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE10 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE10. 000 Select mux mode: ALT0 mux port: GPIO[115] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[23] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON3 of instance: src. 100 Select mux mode: ALT4 mux port: LCD10 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[39] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE10. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE10. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE10. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE10. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE10. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTE10 field descriptions (continued)

Field	Description
	011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE10. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE10. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE10. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE10. 0 Disabled 1 Enabled

5.2.117 Software MUX Pad Control Register 116 (IOMUXC_PTE11)

Address: 4004_8000h base + 1D0h offset = 4004_81D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE11 field descriptions

Field	Description
31–23 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_PTE11 field descriptions (continued)

Field	Description
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE11. 000 Select mux mode: ALT0 mux port: GPIO[116] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[24] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON4 of instance: src. 100 Select mux mode: ALT4 mux port: LCD11 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[40] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE11. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE11. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE11. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE11. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE11. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE11. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled

Table continues on the next page...

IOMUXC_PTE11 field descriptions (continued)

Field	Description
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE11. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE11. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE11. 0 Disabled 1 Enabled

5.2.118 Software MUX Pad Control Register 117 (IOMUXC_PTE12)

Address: 4004_8000h base + 1D4h offset = 4004_81D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE12 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTE12. 000 Select mux mode: ALT0 mux port: GPIO[117] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[25] of instance: tcon0. 010 Select mux mode: ALT2 mux port: CS3 of instance: dspi1. 011 Select mux mode: ALT3 mux port: RCON5 of instance: src. 100 Select mux mode: ALT4 mux port: LCD12 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: LP_IN of instance: lptimer.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE12. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)

Table continues on the next page...

IOMUXC_PTE12 field descriptions (continued)

Field	Description
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE12. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE12. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE12. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE12. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE12. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE12. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE12. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE12. 0 Disabled 1 Enabled

5.2.119 Software MUX Pad Control Register 118 (IOMUXC_PTE13)

Address: 4004_8000h base + 1D8h offset = 4004_81D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE13 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE13. 000 Select mux mode: ALT0 mux port: GPIO[118] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[10] of instance: tcon0. 100 Select mux mode: ALT4 mux port: LCD13 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[41] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE13. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE13. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE13. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE13. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE13. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTE13 field descriptions (continued)

Field	Description
	100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE13. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE13. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE13. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE13. 0 Disabled 1 Enabled

5.2.120 Software MUX Pad Control Register 119 (IOMUXC_PTE14)

Address: 4004_8000h base + 1DCh offset = 4004_81DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE14 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE14.

Table continues on the next page...

IOMUXC_PTE14 field descriptions (continued)

Field	Description
	000 Select mux mode: ALT0 mux port: GPIO[119] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[11] of instance: tcon0. 100 Select mux mode: ALT4 mux port: LCD14 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[42] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE14. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE14. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE14. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE14. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE14. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE14. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE14. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE14. 0 Keeper enable 1 Pull enable

Table continues on the next page...

IOMUXC_PTE14 field descriptions (continued)

Field	Description
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE14. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE14. 0 Disabled 1 Enabled

5.2.121 Software MUX Pad Control Register 120 (IOMUXC_PTE15)

Address: 4004_8000h base + 1E0h offset = 4004_81E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE15 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE15. 000 Select mux mode: ALT0 mux port: GPIO[120] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[12] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON6 of instance: src. 100 Select mux mode: ALT4 mux port: LCD15 of instance: lcd_64f6b. 111 Select mux mode: ALT7 mux port: debug_out[43] of instance: viu_mux.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE15. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE15. 0 Slow Slew Rate 1 Fast Slew Rate

Table continues on the next page...

IOMUXC_PTE15 field descriptions (continued)

Field	Description
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE15. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE15. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE15. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE15. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE15. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE15. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE15. 0 Disabled 1 Enabled

5.2.122 Software MUX Pad Control Register 121 (IOMUXC_PTE16)

Address: 4004_8000h base + 1E4h offset = 4004_81E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE16 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE16. 000 Select mux mode: ALT0 mux port: GPIO[121] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[13] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON7 of instance: src. 100 Select mux mode: ALT4 mux port: LCD16 of instance: lcd_64f6b.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE16. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE16. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE16. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE16. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE16. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTE16 field descriptions (continued)

Field	Description
	100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE16. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE16. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE16. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE16. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE16. 0 Disabled 1 Enabled

5.2.123 Software MUX Pad Control Register 122 (IOMUXC_PTE17)

Address: 4004_8000h base + 1E8h offset = 4004_81E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE17 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE17.

Table continues on the next page...

IOMUXC_PTE17 field descriptions (continued)

Field	Description
	000 Select mux mode: ALT0 mux port: GPIO[122] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[14] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON8 of instance: src. 100 Select mux mode: ALT4 mux port: LCD17 of instance: lcd_64f6b.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE17. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE17. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE17. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE17. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE17. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE17. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE17. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE17. 0 Keeper enable 1 Pull enable

Table continues on the next page...

IOMUXC_PTE17 field descriptions (continued)

Field	Description
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE17. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE17. 0 Disabled 1 Enabled

5.2.124 Software MUX Pad Control Register 123 (IOMUXC_PTE18)

Address: 4004_8000h base + 1ECh offset = 4004_81ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE18 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE18. 000 Select mux mode: ALT0 mux port: GPIO[123] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[15] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON9 of instance: src. 100 Select mux mode: ALT4 mux port: LCD18 of instance: lcd_64f6b.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE18. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE18. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE18.

Table continues on the next page...

IOMUXC_PTE18 field descriptions (continued)

Field	Description
	0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE18. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE18. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE18. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE18. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE18. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE18. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE18. 0 Disabled 1 Enabled

5.2.125 Software MUX Pad Control Register 124 (IOMUXC_PTE19)

Address: 4004_8000h base + 1F0h offset = 4004_81F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE19 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE19. **NOTE:** Pad PTE19 is involved in Daisy Chain. • Config Register IOMUXC_I2C0_IPP_SCL_IND_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[124] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[16] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON10 of instance: src. 100 Select mux mode: ALT4 mux port: LCD19 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: SCL of instance: i2c0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE19. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE19. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE19. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE19. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE19. 000 output driver disabled;

Table continues on the next page...

IOMUXC_PTE19 field descriptions (continued)

Field	Description
	001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE19. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE19. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE19. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE19. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE19. 0 Disabled 1 Enabled

5.2.126 Software MUX Pad Control Register 125 (IOMUXC_PTE20)

Address: 4004_8000h base + 1F4h offset = 4004_81F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE20 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTE20. **NOTE:** Pad PTE20 is involved in Daisy Chain. • Config Register IOMUXC_I2C0_IPP_SDA_IND_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[125] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[17] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON11 of instance: src. 100 Select mux mode: ALT4 mux port: LCD20 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: SDA of instance: i2c0. 111 Select mux mode: ALT7 mux port: in of instance: ewm.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE20. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE20. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE20. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE20. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE20. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE20. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up

Table continues on the next page...

IOMUXC_PTE20 field descriptions (continued)

Field	Description
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE20. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE20. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE20. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE20. 0 Disabled 1 Enabled

5.2.127 Software MUX Pad Control Register 126 (IOMUXC_PTE21)

Address: 4004_8000h base + 1F8h offset = 4004_81F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE21 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 3 IOMUX modes to be used for pad: PTE21. 000 Select mux mode: ALT0 mux port: GPIO[126] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[2] of instance: tcon0. 100 Select mux mode: ALT4 mux port: LCD21 of instance: lcd_64f6b.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE21. 00 Low (50 MHz) 01 Medium (100 MHz)

Table continues on the next page...

IOMUXC_PTE21 field descriptions (continued)

Field	Description
	10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE21. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE21. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE21. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE21. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE21. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE21. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE21. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE21. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE21. 0 Disabled 1 Enabled

5.2.128 Software MUX Pad Control Register 127 (IOMUXC_PTE22)

Address: 4004_8000h base + 1FCh offset = 4004_81FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE22 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 3 IOMUX modes to be used for pad: PTE22. 000 Select mux mode: ALT0 mux port: GPIO[127] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[3] of instance: tcon0. 100 Select mux mode: ALT4 mux port: LCD22 of instance: lcd_64f6b.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE22. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE22. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE22. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE22. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE22. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTE22 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE22. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE22. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE22. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE22. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE22. 0 Disabled 1 Enabled

5.2.129 Software MUX Pad Control Register 128 (IOMUXC_PTE23)

Address: 4004_8000h base + 200h offset = 4004_8200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE23 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE23. 000 Select mux mode: ALT0 mux port: GPIO[128] of instance: rgpioc.

Table continues on the next page...

IOMUXC_PTE23 field descriptions (continued)

Field	Description
	001 Select mux mode: ALT1 mux port: DATA_OUT[4] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON12 of instance: src. 100 Select mux mode: ALT4 mux port: LCD23 of instance: lcd_64f6b.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE23. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE23. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE23. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE23. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE23. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE23. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE23. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE23. 0 Keeper enable 1 Pull enable

Table continues on the next page...

IOMUXC_PTE23 field descriptions (continued)

Field	Description
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE23. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE23. 0 Disabled 1 Enabled

5.2.130 Software MUX Pad Control Register 129 (IOMUXC_PTE24)

Address: 4004_8000h base + 204h offset = 4004_8204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE24 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE24. 000 Select mux mode: ALT0 mux port: GPIO[129] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[5] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON13 of instance: src. 100 Select mux mode: ALT4 mux port: LCD24 of instance: lcd_64f6b.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE24. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE24. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE24.

Table continues on the next page...

IOMUXC_PTE24 field descriptions (continued)

Field	Description
	0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE24. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE24. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE24. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE24. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE24. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE24. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE24. 0 Disabled 1 Enabled

5.2.131 Software MUX Pad Control Register 130 (IOMUXC_PTE25)

Address: 4004_8000h base + 208h offset = 4004_8208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE25 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE25. 000 Select mux mode: ALT0 mux port: GPIO[130] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[6] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON14 of instance: src. 100 Select mux mode: ALT4 mux port: LCD25 of instance: lcd_64f6b.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE25. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE25. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE25. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE25. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE25. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_PTE25 field descriptions (continued)

Field	Description
	100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE25. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE25. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE25. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE25. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE25. 0 Disabled 1 Enabled

5.2.132 Software MUX Pad Control Register 131 (IOMUXC_PTE26)

Address: 4004_8000h base + 20Ch offset = 4004_820Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE26 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 4 IOMUX modes to be used for pad: PTE26.

Table continues on the next page...

IOMUXC_PTE26 field descriptions (continued)

Field	Description
	000 Select mux mode: ALT0 mux port: GPIO[131] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[7] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON15 of instance: src. 100 Select mux mode: ALT4 mux port: LCD26 of instance: lcd_64f6b.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE26. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE26. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE26. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE26. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE26. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE26. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE26. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE26. 0 Keeper enable 1 Pull enable

Table continues on the next page...

IOMUXC_PTE26 field descriptions (continued)

Field	Description
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE26. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE26. 0 Disabled 1 Enabled

5.2.133 Software MUX Pad Control Register 132 (IOMUXC_PTE27)

Address: 4004_8000h base + 210h offset = 4004_8210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE27 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 5 IOMUX modes to be used for pad: PTE27. **NOTE:** Pad PTE27 is involved in Daisy Chain. • Config Register IOMUXC_I2C1_IPP_SCL_IND_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[132] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[8] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON16 of instance: src. 100 Select mux mode: ALT4 mux port: LCD27 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: SCL of instance: i2c1.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE27. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE27.

Table continues on the next page...

IOMUXC_PTE27 field descriptions (continued)

Field	Description
	0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE27. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE27. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE27. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE27. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE27. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE27. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE27. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE27. 0 Disabled 1 Enabled

5.2.134 Software MUX Pad Control Register 133 (IOMUXC_PTE28)

Address: 4004_8000h base + 214h offset = 4004_8214h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved									MUX_MODE			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTE28 field descriptions

Field	Description
31–23 Reserved	This field is reserved.
22–20 MUX_MODE	MUX Mode Select Field. Select 1 of 6 IOMUX modes to be used for pad: PTE28. **NOTE:** Pad PTE28 is involved in Daisy Chain. • Config Register IOMUXC_I2C1_IPP_SDA_IND_SELECT_INPUT for mode ALT5. 000 Select mux mode: ALT0 mux port: GPIO[133] of instance: rgpioc. 001 Select mux mode: ALT1 mux port: DATA_OUT[9] of instance: tcon0. 011 Select mux mode: ALT3 mux port: RCON17 of instance: src. 100 Select mux mode: ALT4 mux port: LCD28 of instance: lcd_64f6b. 101 Select mux mode: ALT5 mux port: SDA of instance: i2c1. 111 Select mux mode: ALT7 mux port: out of instance: ewm.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTE28. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTE28. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTE28. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTE28. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTE28.

Table continues on the next page...

IOMUXC_PTE28 field descriptions (continued)

Field	Description
	000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTE28. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTE28. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTE28. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTE28. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTE28. 0 Disabled 1 Enabled

5.2.135 Software MUX Pad Control Register 134 (IOMUXC_PTA7)

Address: 4004_8000h base + 218h offset = 4004_8218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved											MUX_MODE	Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved		SPEED		SRE	ODE	HYS	DSE			PUS		PKE	PUE	OBE	IBE
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_PTA7 field descriptions

Field	Description
31–21 Reserved	This field is reserved.
20 MUX_MODE	MUX Mode Select Field. Select 1 of 2 IOMUX modes to be used for pad: PTA7. **NOTE:** Pad PTA7 is involved in Daisy Chain. • Config Register IOMUXC_VIDEO_IN0_IPP_IND_PIX_CLK_SELECT_INPUT for mode ALT1. 00 Select mux mode: ALT0 mux port: GPIO[134] of instance: rgpioc. 01 Select mux mode: ALT1 mux port: PIX_CLK of instance: video_in0.
19–14 Reserved	This field is reserved.
13–12 SPEED	Speed Field. Select one of the following values for pad: PTA7. 00 Low (50 MHz) 01 Medium (100 MHz) 10 Medium (100 MHz) 11 High (200 MHz)
11 SRE	Slew Rate Field. Select one of the following values for pad: PTA7. 0 Slow Slew Rate 1 Fast Slew Rate
10 ODE	Open Drain Enable Field. Select one of the following values for pad: PTA7. 0 Output is CMOS 1 Output is open drain
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: PTA7.

Table continues on the next page...

IOMUXC_PTA7 field descriptions (continued)

Field	Description
	0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: PTA7. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: PTA7. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: PTA7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: PTA7. 0 Keeper enable 1 Pull enable
1 OBE	Output Buffer Enable Field. Select one of the following values for pad: PTA7. 0 Disabled 1 Enabled
0 IBE	Input Buffer Enable Field. Select one of the following values for pad: PTA7. 0 Disabled 1 Enabled

5.2.136 Software MUX DDR RESET Pad Configuration Register (IOMUXC_DDR_RESETB)

Address: 4004_8000h base + 21Ch offset = 4004_821Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_RESETB field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_RESETB. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_RESETB. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_RESETB. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_RESETB. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_RESETB field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_RESETB. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_RESETB. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_RESETB. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.137 Software MUX DDR A15 Pad Control Register (IOMUXC_DDR_A_15)

Address: 4004_8000h base + 220h offset = 4004_8220h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_15 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_A_15 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_15. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_15. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_15. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_15. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_15. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_15. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.138 Software MUX DDR A14 Pad Control Register (IOMUXC_DDR_A_14)

Address: 4004_8000h base + 224h offset = 4004_8224h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_14 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_14. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_14. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_14. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_14. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_A_14 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_14. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_14. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_14. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.139 Software MUX DDR A13 Pad Control Register (IOMUXC_DDR_A_13)

Address: 4004_8000h base + 228h offset = 4004_8228h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_13 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_A_13 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_13. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_13. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_13. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_13. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_13. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_13. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.140 Software MUX DDR A12 Pad Control Register (IOMUXC_DDR_A_12)

Address: 4004_8000h base + 22Ch offset = 4004_822Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_12 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_12. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_12. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_12. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_12. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_A_12 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_12. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_12. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.141 Software MUX DDR A11 Pad Control Register (IOMUXC_DDR_A_11)

Address: 4004_8000h base + 230h offset = 4004_8230h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_11 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_A_11 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_11. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_11. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_11. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_11. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_11. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_11. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.142 Software MUX DDR A10 Pad Control Register (IOMUXC_DDR_A_10)

Address: 4004_8000h base + 234h offset = 4004_8234h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_10 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_10. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_10. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_10. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_10. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_A_10 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_10. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_10. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.143 Software MUX DDR A9 Pad Control Register (IOMUXC_DDR_A_9)

Address: 4004_8000h base + 238h offset = 4004_8238h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_9 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_A_9 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_9. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_9. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_9. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_9. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_9. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_9. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.144 Software MUX DDR A8 Pad Control Register (IOMUXC_DDR_A_8)

Address: 4004_8000h base + 23Ch offset = 4004_823Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_8 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_8. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_8. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_8. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_8. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_A_8 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_8. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_8. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.145 Software MUX DDR A7 Pad Control Register (IOMUXC_DDR_A_7)

Address: 4004_8000h base + 240h offset = 4004_8240h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_7 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_A_7 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_7. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_7. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_7. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_7. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_7. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_7. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.146 Software MUX DDR A6 Pad Control Register (IOMUXC_DDR_A_6)

Address: 4004_8000h base + 244h offset = 4004_8244h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_6 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_6. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_6. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_6. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_6. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_A_6 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_6. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_6. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.147 Software MUX DDR A5 Pad Control Register (IOMUXC_DDR_A_5)

Address: 4004_8000h base + 248h offset = 4004_8248h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_5 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_A_5 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_5. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_5. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_5. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_5. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_5. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_5. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.148 Software MUX DDR A4 Pad Control Register (IOMUXC_DDR_A_4)

Address: 4004_8000h base + 24Ch offset = 4004_824Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_4 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_4. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_4. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_4. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_4. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_A_4 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_4. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_4. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.149 Software MUX DDR Pad A3 Control Register (IOMUXC_DDR_A_3)

Address: 4004_8000h base + 250h offset = 4004_8250h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_3 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_A_3 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_3. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_3. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_3. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_3. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_3. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_3. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.150 Software MUX DDR A2 Pad Control Register (IOMUXC_DDR_A_2)

Address: 4004_8000h base + 254h offset = 4004_8254h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_2 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_2. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_2. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_2. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_2. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_A_2 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_2. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_2. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.151 Software MUX DDR A1 Pad Control Register (IOMUXC_DDR_A_1)

Address: 4004_8000h base + 258h offset = 4004_8258h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_1 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_A_1 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_1. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_1. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_1. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.152 Software MUX DDR A0 Pad Control Register (IOMUXC_DDR_A_0)

Address: 4004_8000h base + 25Ch offset = 4004_825Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_A_0 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_A_0. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_A_0. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_A_0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_A_0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_A_0 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_A_0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_A_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_A_0. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.153 Software MUX DDR BA2 Pad Control Register (IOMUXC_DDR_BA_2)

Address: 4004_8000h base + 260h offset = 4004_8260h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_BA_2 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_BA_2 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_BA_2. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_BA_2. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_BA_2. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_BA_2. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_BA_2. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_BA_2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_BA_2. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.154 Software MUX DDR BA1 Pad Control Register (IOMUXC_DDR_BA_1)

Address: 4004_8000h base + 264h offset = 4004_8264h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_BA_1 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_BA_1. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_BA_1. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_BA_1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_BA_1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_BA_1 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_BA_1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_BA_1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_BA_1. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.155 Software MUX DDR BA0 Pad Control Register (IOMUXC_DDR_BA_0)

Address: 4004_8000h base + 268h offset = 4004_8268h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_BA_0 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_BA_0 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_BA_0. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_BA_0. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_BA_0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_BA_0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_BA_0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_BA_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_BA_0. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.156 Software MUX DDR CAS Pad Control Register (IOMUXC_DDR_CAS_B)

Address: 4004_8000h base + 26Ch offset = 4004_826Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CAS_B field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_CAS_b. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_CAS_b. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_CAS_b. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_CAS_b. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CAS_B field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_CAS_b. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_CAS_b. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_CAS_b. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.157 Software MUX DDR CKE0 Pad Control Register (IOMUXC_DDR_CKE_0)

Address: 4004_8000h base + 270h offset = 4004_8270h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0

IOMUXC_DDR_CKE_0 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CKE_0 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_CKE_0. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_CKE_0. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_CKE_0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_CKE_0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_CKE_0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_CKE_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_CKE_0. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.158 Software MUX DDR CLK0 Pad Control Register (IOMUXC_DDR_CLK_0)

Address: 4004_8000h base + 274h offset = 4004_8274h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CLK_0 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_CLK_0. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_CLK_0. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_CLK_0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_CLK_0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CLK_0 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_CLK_0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_CLK_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_CLK_0. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.159 Software MUX DDR CS B0 Pad Control Register (IOMUXC_DDR_CS_B_0)

Address: 4004_8000h base + 278h offset = 4004_8278h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_B_0 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CS_B_0 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_CS_b_0. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_CS_b_0. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_CS_b_0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_CS_b_0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_CS_b_0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_CS_b_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_CS_b_0. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.160 Software MUX DDR CS D15 Pad Control Register (IOMUXC_DDR_CS_D_15)

Address: 4004_8000h base + 27Ch offset = 4004_827Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_15 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_15. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_15. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_15. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_15. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CS_D_15 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_15. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_15. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_15. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.161 Software MUX DDR CS D14 Pad Control Register (IOMUXC_DDR_CS_D_14)

Address: 4004_8000h base + 280h offset = 4004_8280h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_14 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CS_D_14 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_14. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_14. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_14. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_14. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_14. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_14. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_14. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.162 Software MUX DDR CS D13 Pad Control Register (IOMUXC_DDR_CS_D_13)

Address: 4004_8000h base + 284h offset = 4004_8284h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_13 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_13. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_13. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_13. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_13. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CS_D_13 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_13. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_13. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_13. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.163 Software MUX DDR CS D12 Pad Control Register (IOMUXC_DDR_CS_D_12)

Address: 4004_8000h base + 288h offset = 4004_8288h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_12 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CS_D_12 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_12. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_12. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_12. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_12. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_12. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_12. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_12. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.164 Software MUX DDR CS D11 Pad Control Register (IOMUXC_DDR_CS_D_11)

Address: 4004_8000h base + 28Ch offset = 4004_828Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_11 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_11. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_11. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_11. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_11. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CS_D_11 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_11. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_11. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_11. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.165 Software MUX DDR CS D10 Pad Control Register (IOMUXC_DDR_CS_D_10)

Address: 4004_8000h base + 290h offset = 4004_8290h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_10 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CS_D_10 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_10. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_10. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_10. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_10. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_10. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_10. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_10. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.166 Software MUX DDR CS D9 Pad Control Register (IOMUXC_DDR_CS_D_9)

Address: 4004_8000h base + 294h offset = 4004_8294h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_9 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_9. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_9. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_9. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_9. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CS_D_9 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_9. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_9. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_9. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.167 Software MUX DDR CS D8 Pad Control Register (IOMUXC_DDR_CS_D_8)

Address: 4004_8000h base + 298h offset = 4004_8298h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_8 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CS_D_8 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_8. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_8. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_8. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_8. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_8. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_8. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_8. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.168 Software MUX DDR CS D7 Pad Control Register (IOMUXC_DDR_CS_D_7)

Address: 4004_8000h base + 29Ch offset = 4004_829Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_7 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_7. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_7. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_7. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_7. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CS_D_7 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_7. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_7. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_7. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.169 Software MUX DDR CS D6 Pad Control Register (IOMUXC_DDR_CS_D_6)

Address: 4004_8000h base + 2A0h offset = 4004_82A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_6 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CS_D_6 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_6. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_6. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_6. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_6. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_6. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_6. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_6. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.170 Software MUX DDR CS D5 Pad Control Register (IOMUXC_DDR_CS_D_5)

Address: 4004_8000h base + 2A4h offset = 4004_82A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_5 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_5. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_5. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_5. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_5. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CS_D_5 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_5. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_5. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_5. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.171 Software MUX DDR CS D4 Pad Control Register (IOMUXC_DDR_CS_D_4)

Address: 4004_8000h base + 2A8h offset = 4004_82A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_4 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CS_D_4 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_4. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_4. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_4. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_4. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_4. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_4. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_4. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.172 Software MUX DDR CS D3 Pad Control Register (IOMUXC_DDR_CS_D_3)

Address: 4004_8000h base + 2ACh offset = 4004_82ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_3 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_3. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_3. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_3. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_3. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CS_D_3 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_3. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_3. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_3. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.173 Software MUX DDR CS D2 Pad Control Register (IOMUXC_DDR_CS_D_2)

Address: 4004_8000h base + 2B0h offset = 4004_82B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_2 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CS_D_2 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_2. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_2. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_2. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_2. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_2. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_2. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.174 Software MUX DDR CS D1 Pad Control Register (IOMUXC_DDR_CS_D_1)

Address: 4004_8000h base + 2B4h offset = 4004_82B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_1 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_1. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_1. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_CS_D_1 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_1. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.175 Software MUX DDR CS D0 Pad Control Register (IOMUXC_DDR_CS_D_0)

Address: 4004_8000h base + 2B8h offset = 4004_82B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_CS_D_0 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_CS_D_0 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_D_0. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_D_0. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_D_0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_D_0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_D_0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_D_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_D_0. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.176 Software MUX DDR DQM1 Pad Control Register (IOMUXC_DDR_DQM_1)

Address: 4004_8000h base + 2BCh offset = 4004_82BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_DQM_1 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_DQM_1. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_DQM_1. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_DQM_1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_DQM_1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_DQM_1 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_DQM_1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_DQM_1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_DQM_1. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.177 Software MUX DDR DQM0 Pad Control Register 0 (IOMUXC_DDR_DQM_0)

Address: 4004_8000h base + 2C0h offset = 4004_82C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_DQM_0 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_DQM_0 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_DQM_0. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_DQM_0. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_DQM_0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_DQM_0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_DQM_0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_DQM_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_DQM_0. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.178 Software MUX DDR DQS1 Pad Control Register 1 (IOMUXC_DDR_DQS_1)

Address: 4004_8000h base + 2C4h offset = 4004_82C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_DQS_1 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_DQS_1. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_DQS_1. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_DQS_1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_DQS_1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_DQS_1 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_DQS_1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_DQS_1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_DQS_1. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.179 Software MUX DDR DQS0 Pad Control Register 0 (IOMUXC_DDR_DQS_0)

Address: 4004_8000h base + 2C8h offset = 4004_82C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_DQS_0 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_DQS_0 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_DQS_0. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_DQS_0. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_DQS_0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_DQS_0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_DQS_0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_DQS_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_DQS_0. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.180 Software MUX DDR RAS Pad Control Register (IOMUXC_DDR_RAS_B)

Address: 4004_8000h base + 2CCh offset = 4004_82CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_RAS_B field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_RAS_b. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_RAS_b. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_RAS_b. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_RAS_b. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_RAS_B field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_RAS_b. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_RAS_b. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_RAS_b. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.181 Software MUX DDR WE Pad Control Register (IOMUXC_DDR_WE_B)

Address: 4004_8000h base + 2D0h offset = 4004_82D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_WE_B field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_WE_B field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_WE_b. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_WE_b. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_WE_b. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_WE_b. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_WE_b. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_WE_b. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_WE_b. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.182 Software MUX DDR ODT0 Pad Control Register (IOMUXC_DDR_ODT_0)

Address: 4004_8000h base + 2D4h offset = 4004_82D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_ODT_0 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_ODT_0. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_ODT_0. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_ODT_0. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_ODT_0. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DDR_ODT_0 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_ODT_0. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_ODT_0. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_ODT_0. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.183 Software MUX DDR ODT1 Pad Control Register (IOMUXC_DDR_ODT_1)

Address: 4004_8000h base + 2D8h offset = 4004_82D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DDR_ODT_1 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DDR_ODT_1 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DDR_ODT_1. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DDR_ODT_1. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DDR_ODT_1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DDR_ODT_1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DDR_ODT_1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DDR_ODT_1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DDR_ODT_1. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.184 Software MUX Dummy DDRBYTE1 Pad Control Register (IOMUXC_DUMMY_DDRBYTE1)

Address: 4004_8000h base + 2DCh offset = 4004_82DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DUMMY_DDRBYTE1 field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DUMMY_DDRBYTE1. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DUMMY_DDRBYTE1. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DUMMY_DDRBYTE1. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DUMMY_DDRBYTE1. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR)

Table continues on the next page...

IOMUXC_DUMMY_DDRBYTE1 field descriptions (continued)

Field	Description
	101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DUMMY_DDRBYTE1. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DUMMY_DDRBYTE1. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DUMMY_DDRBYTE1. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.185 Software MUX Dummy DDRBYTE2 Pad Control Register (IOMUXC_DUMMY_DDRBYTE2)

Address: 4004_8000h base + 2E0h offset = 4004_82E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															DDR_INPUT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DDR_TRIM		Reserved				HYS	DSE			PUS		PKE	PUE	Reserved	
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

IOMUXC_DUMMY_DDRBYTE2 field descriptions

Field	Description
31–17 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_DUMMY_DDRBYTE2 field descriptions (continued)

Field	Description
16 DDR_INPUT	DDR / CMOS Input Mode Field. Select one of the following values for pad: DUMMY_DDRBYTE2. 0 CMOS input type 1 Differential input mode
15–14 DDR_TRIM	DDR TRIM Field. Select one of the following values for pad: DUMMY_DDRBYTE2. 00 Minimum do->pad delay 01 50 ps do->pad delay 10 100 ps do->pad delay 11 150 ps do->pad delay
13–10 Reserved	This field is reserved.
9 HYS	Hysteresis Enable Field. Select one of the following values for pad: DUMMY_DDRBYTE2. 0 CMOS input 1 Schmitt trigger input
8–6 DSE	Drive Strength Field. Select one of the following values for pad: DUMMY_DDRBYTE2. 000 output driver disabled; 001 150 Ohm (240 Ohm if pad is DDR) 010 75 Ohm (120 Ohm if pad is DDR) 011 50 Ohm (80 Ohm if pad is DDR) 100 37 Ohm (60 Ohm if pad is DDR) 101 30 Ohm (48 Ohm if pad is DDR) 110 25 Ohm 111 20 Ohm (34 Ohm if pad is DDR)
5–4 PUS	Pull Up / Down Config Field. Select one of the following values for pad: DUMMY_DDRBYTE2. 00 100 kOhm Pull Down 01 47 kOhm Pull Up 10 100 kOhm Pull Up 11 22 kOhm Pull Up
3 PKE	Pull / Keep Enable Field. Select one of the following values for pad: DUMMY_DDRBYTE2. 0 Pull/Keeper Disabled 1 Pull/Keeper Enabled
2 PUE	Pull / Keep Select Field. Select one of the following values for pad: DUMMY_DDRBYTE2. 0 Keeper enable 1 Pull enable
1–0 Reserved	This field is reserved.

5.2.186 CCM Audio External Clock Input Select Register (IOMUXC_CCM_AUD_EXT_CLK_SELECT_INPUT)

Address: 4004_8000h base + 2ECh offset = 4004_82ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_AUD_EXT_CLK_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ccm, In Pin: aud_ext_clk 00 Selecting Pad: PTA10 for Mode: ALT2. 01 Selecting Pad: PTA12 for Mode: ALT2. 10 Selecting Pad: PTB18 for Mode: ALT2.

5.2.187 CCM Ethernet External Clock Input Select Register (IOMUXC_CCM_ENET_EXT_CLK_SELECT_INPUT)

Address: 4004_8000h base + 2F0h offset = 4004_82F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_ENET_EXT_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ccm, In Pin: enet_ext_clk 0 Selecting Pad: PTA6 for Mode: ALT2. 1 Selecting Pad: PTA9 for Mode: ALT3.

5.2.188 CCM Ethernet TS Clock Input Select Register (IOMUXC_CCM_ENET_TS_CLK_SELECT_INPUT)

Address: 4004_8000h base + 2F4h offset = 4004_82F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_CCM_ENET_TS_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: ccm, In Pin: enet_ts_clk 0 Selecting Pad: PTA10 for Mode: ALT6. 1 Selecting Pad: PTB10 for Mode: ALT7.

5.2.189 DSPI1 SCK Input Select Register (IOMUXC_DSPI1_IPP_IND_SCK_SELECT_INPUT)

Address: 4004_8000h base + 2F8h offset = 4004_82F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_DSPI1_IPP_IND_SCK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: dspi1, In Pin: ipp_ind_sck 0 Selecting Pad: PTC8 for Mode: ALT3. 1 Selecting Pad: PTD8 for Mode: ALT3.

5.2.190 DSPI1 SIN Input Select Register (IOMUXC_DSPI1_IPP_IND_SIN_SELECT_INPUT)

Address: 4004_8000h base + 2FCh offset = 4004_82FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_DSPI1_IPP_IND_SIN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: dspi1, In Pin: ipp_ind_sin 0 Selecting Pad: PTC6 for Mode: ALT3. 1 Selecting Pad: PTD6 for Mode: ALT3.

5.2.191 DSPI1 SS Input Select Register (IOMUXC_DSPI1_IPP_IND_SS_B_SELECT_INPUT)

Address: 4004_8000h base + 300h offset = 4004_8300h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_DSPI1_IPP_IND_SS_B_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: dspi1, In Pin: ipp_ind_ss_b 0 Selecting Pad: PTC5 for Mode: ALT3. 1 Selecting Pad: PTD5 for Mode: ALT3.

5.2.192 Ethernet MAC0 TIMER0 Input Select Register (IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_0_SELECT_INPUT)

Address: 4004_8000h base + 304h offset = 4004_8304h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet_swiahb, In Pin: ipp_ind_mac0_timer[0] 0 Selecting Pad: PTB11 for Mode: ALT7. 1 Selecting Pad: PTD23 for Mode: ALT4.

5.2.193 Ethernet MAC0 TIMER1 Input Select Register (IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_1_SELECT_INPUT)

Address: 4004_8000h base + 308h offset = 4004_8308h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ENET_SWIAHB_IPP_IND_MAC0_TIMER_1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: enet_swiahb, In Pin: ipp_ind_mac0_timer[1] 0 Selecting Pad: PTB12 for Mode: ALT7. 1 Selecting Pad: PTD22 for Mode: ALT4.

5.2.194 ESAI FST Input Select Register (IOMUXC_ESAI_IPP_IND_FST_SELECT_INPUT)

Address: 4004_8000h base + 30Ch offset = 4004_830Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_IPP_IND_FST_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai, In Pin: ipp_ind_fst 0 Selecting Pad: PTC1 for Mode: ALT4. 1 Selecting Pad: PTC10 for Mode: ALT3.

5.2.195 ESAI SCKT Input Select Register (IOMUXC_ESAI_IPP_IND_SCKT_SELECT_INPUT)

Address: 4004_8000h base + 310h offset = 4004_8310h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_IPP_IND_SCKT_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai, In Pin: ipp_ind_sckt 0 Selecting Pad: PTC0 for Mode: ALT4. 1 Selecting Pad: PTC9 for Mode: ALT3.

5.2.196 ESAI SDO0 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO0_SELECT_INPUT)

Address: 4004_8000h base + 314h offset = 4004_8314h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_IPP_IND_SDO0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai, In Pin: ipp_ind_sdo0 0 Selecting Pad: PTC2 for Mode: ALT4. 1 Selecting Pad: PTC11 for Mode: ALT3.

5.2.197 ESAI SDO1 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO1_SELECT_INPUT)

Address: 4004_8000h base + 318h offset = 4004_8318h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_IPP_IND_SDO1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai, In Pin: ipp_ind_sdo1 0 Selecting Pad: PTC3 for Mode: ALT4. 1 Selecting Pad: PTC12 for Mode: ALT3.

5.2.198 ESAI SDO2 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO2_SDI3_SELECT_INPUT)

Address: 4004_8000h base + 31Ch offset = 4004_831Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_IPP_IND_SDO2_SDI3_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai, In Pin: ipp_ind_sdo2_sdi3 0 Selecting Pad: PTC4 for Mode: ALT4. 1 Selecting Pad: PTC13 for Mode: ALT3.

5.2.199 ESAI SDO3 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO3_SDI2_SELECT_INPUT)

Address: 4004_8000h base + 320h offset = 4004_8320h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_IPP_IND_SDO3_SDI2_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai, In Pin: ipp_ind_sdo3_sdi2 0 Selecting Pad: PTC5 for Mode: ALT4. 1 Selecting Pad: PTC14 for Mode: ALT3.

5.2.200 ESAI SDO4 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO4_SDI1_SELECT_INPUT)

Address: 4004_8000h base + 324h offset = 4004_8324h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_IPP_IND_SDO4_SDI1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai, In Pin: ipp_ind_sdo4_sdi1 0 Selecting Pad: PTC7 for Mode: ALT4. 1 Selecting Pad: PTC16 for Mode: ALT3.

5.2.201 ESAI SDO5 Input Select Register (IOMUXC_ESAI_IPP_IND_SDO5_SDI0_SELECT_INPUT)

Address: 4004_8000h base + 328h offset = 4004_8328h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_ESAI_IPP_IND_SDO5_SDI0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: esai, In Pin: ipp_ind_sdo5_sdi0 0 Selecting Pad: PTC6 for Mode: ALT4. 1 Selecting Pad: PTC15 for Mode: ALT3.

5.2.202 FlexTimer1 CH0 Input Select Register (IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_0_SELECT_INPUT)

Address: 4004_8000h base + 32Ch offset = 4004_832Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_0_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flextimer1, In Pin: ipp_ind_ftm_ch[0] 0 Selecting Pad: PTB8 for Mode: ALT1. 1 Selecting Pad: PTC0 for Mode: ALT2.

5.2.203 FlexTimer1 CH1 Input Select Register (IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_1_SELECT_INPUT)

Address: 4004_8000h base + 330h offset = 4004_8330h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_FLEXTIMER1_IPP_IND_FTM_CH_1_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flextimer1, In Pin: ipp_ind_ftm_ch[1] 0 Selecting Pad: PTB9 for Mode: ALT1. 1 Selecting Pad: PTC1 for Mode: ALT2.

5.2.204 FlexTimer1 PHA Input Select Register (IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHA_SELECT_INPUT)

Address: 4004_8000h base + 334h offset = 4004_8334h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHA_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flextimer1, In Pin: ipp_ind_ftm_pha 0 Selecting Pad: PTA18 for Mode: ALT3. 1 Selecting Pad: PTB8 for Mode: ALT3.

5.2.205 FlexTimer1 PHB Input Select Register (IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHB_SELECT_INPUT)

Address: 4004_8000h base + 338h offset = 4004_8338h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_FLEXTIMER1_IPP_IND_FTM_PHB_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: flextimer1, In Pin: ipp_ind_ftm_phb 0 Selecting Pad: PTA19 for Mode: ALT3. 1 Selecting Pad: PTB9 for Mode: ALT3.

5.2.206 I2C0 SCL Input Select Register (IOMUXC_I2C0_IPP_SCL_IND_SELECT_INPUT)

Address: 4004_8000h base + 33Ch offset = 4004_833Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_I2C0_IPP_SCL_IND_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c0, In Pin: ipp_scl_ind 00 Selecting Pad: PTA12 for Mode: ALT7. 01 Selecting Pad: PTB14 for Mode: ALT2. 10 Selecting Pad: PTD19 for Mode: ALT4. 11 Selecting Pad: PTE19 for Mode: ALT5.

5.2.207 I2C0 SDA Input Select Register (IOMUXC_I2C0_IPP_SDA_IND_SELECT_INPUT)

Address: 4004_8000h base + 340h offset = 4004_8340h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_I2C0_IPP_SDA_IND_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c0, In Pin: ipp_sda_ind 00 Selecting Pad: PTA16 for Mode: ALT7. 01 Selecting Pad: PTB15 for Mode: ALT2. 10 Selecting Pad: PTD18 for Mode: ALT4. 11 Selecting Pad: PTE20 for Mode: ALT5.

5.2.208 I2C1 SCL Input Select Register (IOMUXC_I2C1_IPP_SCL_IND_SELECT_INPUT)

Address: 4004_8000h base + 344h offset = 4004_8344h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_I2C1_IPP_SCL_IND_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c1, In Pin: ipp_scl_ind 00 Selecting Pad: PTA17 for Mode: ALT7. 01 Selecting Pad: PTB16 for Mode: ALT2. 10 Selecting Pad: PTD17 for Mode: ALT4. 11 Selecting Pad: PTE27 for Mode: ALT5.

5.2.209 I2C1 SDA Input Select Register (IOMUXC_I2C1_IPP_SDA_IND_SELECT_INPUT)

Address: 4004_8000h base + 348h offset = 4004_8348h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_I2C1_IPP_SDA_IND_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.

Table continues on the next page...

IOMUXC_I2C1_IPP_SDA_IND_SELECT_INPUT field descriptions (continued)

Field	Description
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c1, In Pin: ipp_sda_ind 00 Selecting Pad: PTA18 for Mode: ALT7. 01 Selecting Pad: PTB17 for Mode: ALT2. 10 Selecting Pad: PTD16 for Mode: ALT4. 11 Selecting Pad: PTE28 for Mode: ALT5.

5.2.210 I2C2 SCL Input Select Register (IOMUXC_I2C2_IPP_SCL_IND_SELECT_INPUT)

Address: 4004_8000h base + 34Ch offset = 4004_834Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_I2C2_IPP_SCL_IND_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c2, In Pin: ipp_scl_ind 0 Selecting Pad: PTA22 for Mode: ALT6. 1 Selecting Pad: PTD28 for Mode: ALT3.

5.2.211 I2C2 SDA Input Select Register (IOMUXC_I2C2_IPP_SDA_IND_SELECT_INPUT)

Address: 4004_8000h base + 350h offset = 4004_8350h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_I2C2_IPP_SDA_IND_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: i2c2, In Pin: ipp_sda_ind 0 Selecting Pad: PTA23 for Mode: ALT6. 1 Selecting Pad: PTD27 for Mode: ALT3.

5.2.212 MediaLB Clock Input Select Register (IOMUXC_MLB_TOP_MLBCLK_IN_SELECT_INPUT)

Address: 4004_8000h base + 354h offset = 4004_8354h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_MLB_TOP_MLBCLK_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mlb_top, In Pin: mlbclk_in 0 Selecting Pad: PTA8 for Mode: ALT7. 1 Selecting Pad: PTC9 for Mode: ALT6.

5.2.213 MediaLB Data Input Select Register (IOMUXC_MLB_TOP_MLBDAT_IN_SELECT_INPUT)

Address: 4004_8000h base + 358h offset = 4004_8358h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_MLB_TOP_MLBDAT_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mlb_top, In Pin: mlbdat_in 0 Selecting Pad: PTA11 for Mode: ALT7. 1 Selecting Pad: PTC11 for Mode: ALT6.

5.2.214 MediaLB Signal Input Select Register (IOMUXC_MLB_TOP_MLBSIG_IN_SELECT_INPUT)

Address: 4004_8000h base + 35Ch offset = 4004_835Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															DAISY
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_MLB_TOP_MLBSIG_IN_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: mlb_top, In Pin: mlbsig_in 0 Selecting Pad: PTA10 for Mode: ALT7. 1 Selecting Pad: PTC10 for Mode: ALT6.

5.2.215 SAI1 TXSYNC Input Select Register (IOMUXC_SAI1_IPP_IND_SAI_TXSYNC_SELECT_INPUT)

Address: 4004_8000h base + 360h offset = 4004_8360h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI1_IPP_IND_SAI_TXSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai1, In Pin: ipp_ind_sai_txsync 0 Selecting Pad: PTD9 for Mode: ALT6. 1 Selecting Pad: PTC31 for Mode: ALT1.

5.2.216 SAI2 RXBCLK Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_RXBCLK_SELECT_INPUT)

Address: 4004_8000h base + 364h offset = 4004_8364h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI2_IPP_IND_SAI_RXBCLK_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai2, In Pin: ipp_ind_sai_rxbclk 00 Selecting Pad: PTA21 for Mode: ALT5. 01 Selecting Pad: PTB0 for Mode: ALT5. 10 Selecting Pad: PTC13 for Mode: ALT5.

5.2.217 SAI2 RXDATA0 Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_RXDATA_0_SELECT_INPUT)

Address: 4004_8000h base + 368h offset = 4004_8368h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI2_IPP_IND_SAI_RXDATA_0_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai2, In Pin: ipp_ind_sai_rxdata[0] 00 Selecting Pad: PTA22 for Mode: ALT5. 01 Selecting Pad: PTB1 for Mode: ALT5. 10 Selecting Pad: PTC14 for Mode: ALT5.

5.2.218 SAI2 RXSYNC Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_RXSYNC_SELECT_INPUT)

Address: 4004_8000h base + 36Ch offset = 4004_836Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI2_IPP_IND_SAI_RXSYNC_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai2, In Pin: ipp_ind_sai_rxsync 00 Selecting Pad: PTA23 for Mode: ALT5. 01 Selecting Pad: PTB2 for Mode: ALT5. 10 Selecting Pad: PTC16 for Mode: ALT5.

5.2.219 SAI2 TXBLCK Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_TXBCLK_SELECT_INPUT)

Address: 4004_8000h base + 370h offset = 4004_8370h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI2_IPP_IND_SAI_TXBCLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai2, In Pin: ipp_ind_sai_txbclk 0 Selecting Pad: PTA16 for Mode: ALT5. 1 Selecting Pad: PTC12 for Mode: ALT5.

5.2.220 SAI2 TXSYNC Input Select Register (IOMUXC_SAI2_IPP_IND_SAI_TXSYNC_SELECT_INPUT)

Address: 4004_8000h base + 374h offset = 4004_8374h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI2_IPP_IND_SAI_TXSYNC_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sai2, In Pin: ipp_ind_sai_txsync 0 Selecting Pad: PTA19 for Mode: ALT5. 1 Selecting Pad: PTC17 for Mode: ALT5.

5.2.221 UART FLX1 CTS Input Select Register (IOMUXC_SCI_FLX1_IPP_IND_CTS_B_SELECT_INPUT)

Address: 4004_8000h base + 378h offset = 4004_8378h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SCI_FLX1_IPP_IND_CTS_B_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sci_flx1, In Pin: ipp_ind_cts_b 00 Selecting Pad: PTB7 for Mode: ALT2. 01 Selecting Pad: PTC5 for Mode: ALT2. 10 Selecting Pad: PTB26 for Mode: ALT2.

5.2.222 UART FLX1 RX Input Select Register (IOMUXC_SCI_FLX1_IPP_IND_SCI_RX_SELECT_INPUT)

Address: 4004_8000h base + 37Ch offset = 4004_837Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SCI_FLX1_IPP_IND_SCI_RX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sci_flx1, In Pin: ipp_ind_sci_rx

Table continues on the next page...

IOMUXC_SCI_FLX1_IPP_IND_SCI_RX_SELECT_INPUT field descriptions (continued)

Field	Description
	00 Selecting Pad: PTB5 for Mode: ALT2. 01 Selecting Pad: PTC3 for Mode: ALT2. 10 Selecting Pad: PTB24 for Mode: ALT2.

5.2.223 UART FLX1 TX Input Select Register (IOMUXC_SCI_FLX1_IPP_IND_SCI_TX_SELECT_INPUT)

Address: 4004_8000h base + 380h offset = 4004_8380h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SCI_FLX1_IPP_IND_SCI_TX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sci_flx1, In Pin: ipp_ind_sci_tx 00 Selecting Pad: PTB4 for Mode: ALT2. 01 Selecting Pad: PTC2 for Mode: ALT2. 10 Selecting Pad: PTB23 for Mode: ALT2.

5.2.224 UART FLX2 CTS Input Select Register (IOMUXC_SCI_FLX2_IPP_IND_CTS_B_SELECT_INPUT)

Address: 4004_8000h base + 384h offset = 4004_8384h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SCI_FLX2_IPP_IND_CTS_B_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sci_flx2, In Pin: ipp_ind_cts_b 0 Selecting Pad: PTD20 for Mode: ALT6. 1 Selecting Pad: PTD3 for Mode: ALT2.

5.2.225 UART FLX2 RX Input Select Register (IOMUXC_SCI_FLX2_IPP_IND_SCI_RX_SELECT_INPUT)

Address: 4004_8000h base + 388h offset = 4004_8388h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SCI_FLX2_IPP_IND_SCI_RX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sci_flx2, In Pin: ipp_ind_sci_rx 00 Selecting Pad: PTB7 for Mode: ALT7. 01 Selecting Pad: PTD22 for Mode: ALT6. 10 Selecting Pad: PTD1 for Mode: ALT2.

5.2.226 UART FLX2 TX Input Select Register (IOMUXC_SCI_FLX2_IPP_IND_SCI_TX_SELECT_INPUT)

Address: 4004_8000h base + 38Ch offset = 4004_838Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SCI_FLX2_IPP_IND_SCI_TX_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sci_flx2, In Pin: ipp_ind_sci_tx 00 Selecting Pad: PTB6 for Mode: ALT7. 01 Selecting Pad: PTD23 for Mode: ALT6. 10 Selecting Pad: PTD0 for Mode: ALT2.

5.2.227 UART FLX3 RX Input Select Register (IOMUXC_SCI_FLX3_IPP_IND_SCI_RX_SELECT_INPUT)

Address: 4004_8000h base + 390h offset = 4004_8390h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SCI_FLX3_IPP_IND_SCI_RX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sci_flx3, In Pin: ipp_ind_sci_rx 0 Selecting Pad: PTA21 for Mode: ALT6. 1 Selecting Pad: PTA31 for Mode: ALT7.

5.2.228 UART FLX3 TX Input Select Register (IOMUXC_SCI_FLX3_IPP_IND_SCI_TX_SELECT_INPUT)

Address: 4004_8000h base + 394h offset = 4004_8394h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SCI_FLX3_IPP_IND_SCI_TX_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: sci_flx3, In Pin: ipp_ind_sci_tx 0 Selecting Pad: PTA20 for Mode: ALT6. 1 Selecting Pad: PTA30 for Mode: ALT7.

5.2.229 BOOTCFG18 Input Select Register (IOMUXC_SRC_IPP_BOOT_CFG_18_SELECT_INPUT)

Address: 4004_8000h base + 398h offset = 4004_8398h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SRC_IPP_BOOT_CFG_18_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: src, In Pin: ipp_boot_cfg[18] 0 Selecting Pad: PTC0 for Mode: ALT7. 1 Selecting Pad: PTB23 for Mode: ALT3.

5.2.230 BOOTCFG19 Input Select Register (IOMUXC_SRC_IPP_BOOT_CFG_19_SELECT_INPUT)

Address: 4004_8000h base + 39Ch offset = 4004_839Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SRC_IPP_BOOT_CFG_19_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: src, In Pin: ipp_boot_cfg[19] 0 Selecting Pad: PTC1 for Mode: ALT7. 1 Selecting Pad: PTB24 for Mode: ALT3.

5.2.231 BOOTCFG20 Input Select Register (IOMUXC_SRC_IPP_BOOT_CFG_20_SELECT_INPUT)

Address: 4004_8000h base + 3A0h offset = 4004_83A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SRC_IPP_BOOT_CFG_20_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: src, In Pin: ipp_boot_cfg[20] 0 Selecting Pad: PTC2 for Mode: ALT7. 1 Selecting Pad: PTB25 for Mode: ALT3.

5.2.232 Video Decoder Input Select Register (IOMUXC_VIDEO_IN0_IPP_IND_DE_SELECT_INPUT)

Address: 4004_8000h base + 3A4h offset = 4004_83A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_VIDEO_IN0_IPP_IND_DE_SELECT_INPUT field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1–0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: video_in0, In Pin: ipp_ind_de 00 Selecting Pad: PTB5 for Mode: ALT5. 01 Selecting Pad: PTB8 for Mode: ALT5. 10 Selecting Pad: PTB10 for Mode: ALT5.

5.2.233 Video IN0 Input Select Register (IOMUXC_VIDEO_IN0_IPP_IND_FID_SELECT_INPUT)

Address: 4004_8000h base + 3A8h offset = 4004_83A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_VIDEO_IN0_IPP_IND_FID_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: video_in0, In Pin: ipp_ind_fid 0 Selecting Pad: PTB4 for Mode: ALT5. 1 Selecting Pad: PTB22 for Mode: ALT5.

5.2.234 Video PIXCLK Input Select Register (IOMUXC_VIDEO_IN0_IPP_IND_PIX_CLK_SELECT_INPUT)

Address: 4004_8000h base + 3ACh offset = 4004_83ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_VIDEO_IN0_IPP_IND_PIX_CLK_SELECT_INPUT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 DAISY	Selecting Pads Involved in Daisy Chain. Instance: video_in0, In Pin: ipp_ind_pix_clk 0 Selecting Pad: PTB15 for Mode: ALT7. 1 Selecting Pad: PTA7 for Mode: ALT1.

5.3 Functional Description

The figure below shows the GPIO Pad .

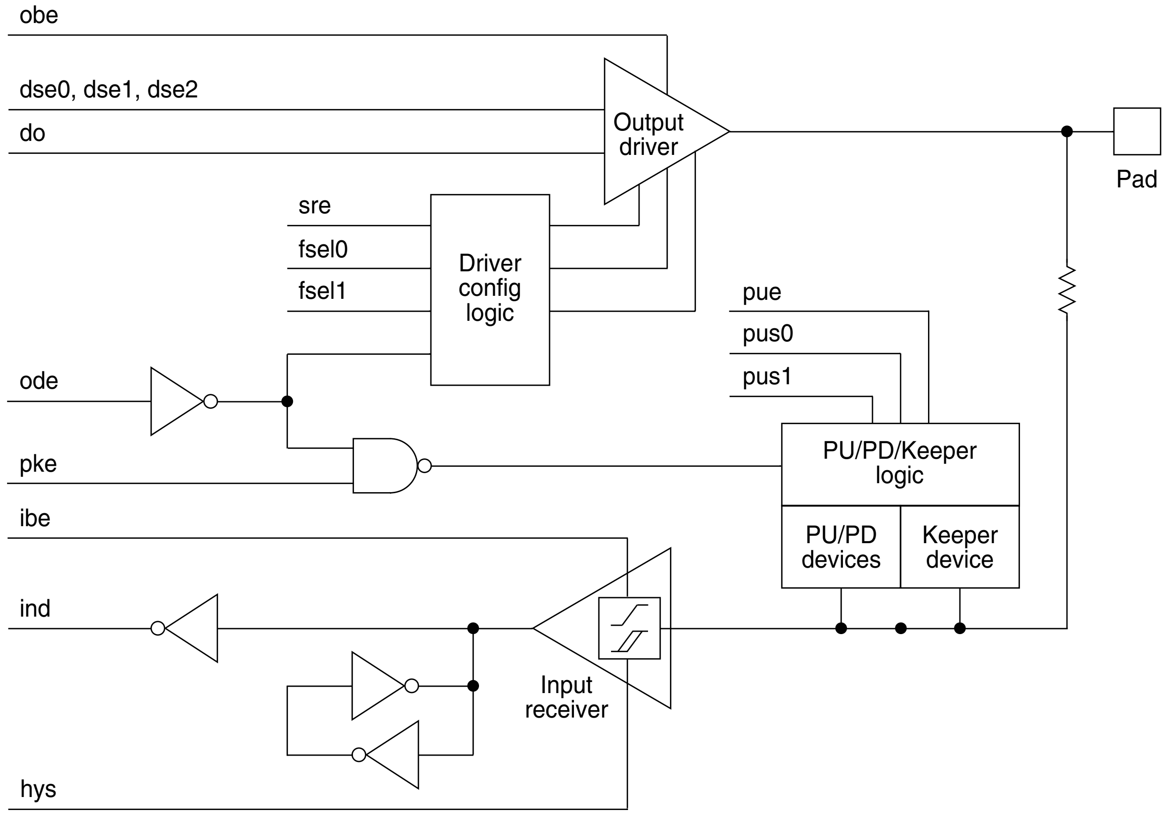


Figure 5-235. GPIO PAD

Table 5-236. Signal Description

Signal Name	Description	Description
Pad	Input/Output	I/O to external world
do	Input	Data coming from the core into the pad
obe	Input	Output enable
ode	Input	Select open drain or CMOS output
dse	Input	Drive Strength select
sre	Input	Slew rate of output buffer
fsel1, fsel0	Input	Slew rate control
pke	Input	Enable pull up, pull down and keeper capability
pue	Input	Enable pull-up/down or keeper
pus	Input	Pull-up/down select
ibe	Input	Input enable
ind	Output	Data coming out of the pad into the core
hys	Input	Select Schmitt trigger or CMOS input

NOTE

For DDR pads, there are additional control signals as explained in IOMUX register map to control DDR functionality.

Table 5-237. Truth Table

ode	obe	pke	pue	pus	do	dse	pad
Output Buffer Mode (functional)							
0	1	0	X	X	do	001, 010, 011, 100, 101, 110, 111	do
0	1	0	X	X	do	0	Z
0	0	0	X	X	X	X	Z
Output Buffer Mode + Keeper							
0	0	1	0	X	X	X	pad (keep previous state)
Output Buffer Mode + pull-up/pull-down resistors							
0	0	1	1	0	X	X	weak0 (pull-down 100 KOhm)
0	0	1	1	1	X	X	weak1 (pull-up 47 KOhm)
0	0	1	1	10	X	X	weak1 (pull-up 100 KOhm)
0	0	1	1	11	X	X	weak1 (pull-up 22 KOhm)
Open Drain Mode							
1	1	0	X	X	do	001, 010, 011, 100, 101, 110, 111	do
0	1	0	X	X	do	0	Z
1	0	0	X	X	X	X	Z
Open Drain Mode + Keeper/Pull-down							
1	1	1	0	X	0	X	0 (open drain + keeper)
1	1	1	0	X	1	X	weak0 (keeper)
1	1	1	1	0	1	X	0 (open drain + pull-down)
Open Drain Mode + Pull-up							
1	1	1	1	1	0	X	0 (open drain + pull-up 47 KOhm)
1	1	1	1	10	0	X	0 (open drain + pull-up 100 KOhm)

Table continues on the next page...

Table 5-237. Truth Table (continued)

ode	obe	pke	pue	pus	do	dse	pad
1	1	1	1	11	0	X	0 (open drain + pull-up 22 KOhm)
1	1	1	1	1	1	X	weak1 (open drain +pull-up 47 KOhm)
1	1	1	1	10	1	X	weak1 (open drain +pull-up 100 KOhm)
1	1	1	1	11	1	X	weak1 (open drain +pull-up 22 KOhm)
1	0	0	X	X	X	X	Z
1	0	1	0	X	X	X	pad (keep previous state)

Table 5-238. Truth Table - Pad to Core

ibe	pad	ind (keep previous state)
0	X	ind
1	pad	pad

IOMUX module is used to configure the Pad settings through the IOMUX registers and multiplex multiple modules on to a single PAD and drive input to a module from multiple PADs. Muxing is done only on obe, ibe, ind and do and rest of the pad settings are through IOMUX registers. DDR pads are dedicated pads for DDR and there is no muxing on them.

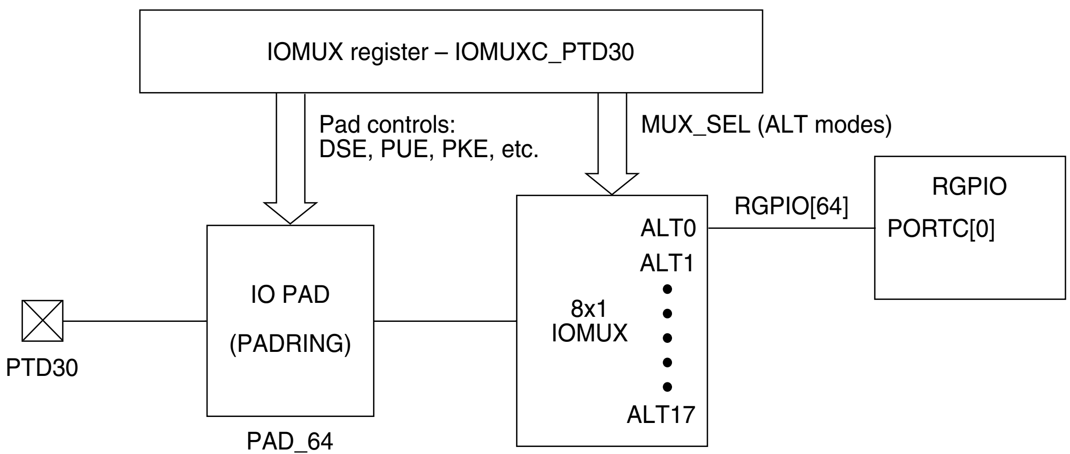


Figure 5-236. Path from the pint PTD30 to RGPIO

5.4 Special Pad Settings

This section describes the special pad setting requirements for several modules in this device. DDR Pins are dedicated pins and do not have any alternate functionality on them. The settings of DDR Pins are controlled by IOMUX as well as DDR module. GPIO Pin settings (speed, sre, ode, hys, dse, pus, pke,pue) are controlled by IOMUX registers, however, special care needs to be taken for ibe/obe. The input buffer of the Pin is enabled by ibe bit in IOMUX register. The output buffer of the Pin is enabled by obe bit of IOMUX only if ALT Mode is configured as GPIO. In other ALT modes, the module configured for ALT mode drives the obe signal to output buffer of Pin and obe bit of IOMUX will have no effect. This is done to reduce the propagation delay it takes for module to enable ibe and data from pin to reach module. This helps to ease timing requirements for modules which enable/disable input buffer dynamically.

5.4.1 DUMMY PADS (DDR/QuadSPI)

There are two dummy pads that are useful for timing calibration of DDR. These pads are internal only, but there corresponding IOMUX register need to be programmed for correct operation of DDR. These registers are:

- IOMUXC_DUMMY_DDRBYTE1 (0x400482DC)
- IOMUXC_DUMMY_DDRBYTE2 (0x400482E0)

DDR: Dummy pads for DDR must be configured before any DDR I/O transactions are done. These pads simulate the input delay of the I/O buffers from the DRAM devices and DDR configures the delays accordingly.

QuadSPI: Dummy pads for QuadSPI can be used to loopback clock while interfacing with devices that do not provide external DQS.

5.4.2 SDHC

Although SDHC_DCLK is output-only signal, ibe needs to be enabled as the module requires output clock to be looped-back through the pad’s input buffer to ease timing requirements.

5.4.3 I2C

I2C pins, SCL and SDA are both Open-drain and bi-directional. Hence, ODE bit should be set in IOMUX registers and ibe enabled.

5.4.4 FlexBus

Flexbus module provides CSPMCR register for additional muxing apart from IOMUX. These registers may need to be programmed depending on the use-case desired.

5.4.5 LCD/ADC

When a pad is configured for LCD or ADC by programming appropriate ALT_MODE, ensure that both ibe/obe of the pad are disabled and all pulls/keeper are disabled as well. This is because input and output buffer are for digital inputs and analog signals from/to LCD/ADC are connected through a resistor directly to pad. If ibe/obe/pke are set, then there will be leakage and potential performance degradation.

5.4.6 SCI

Pull-ups should be enabled to ensure there are no spurious transitions.

5.4.7 Reset Pin Configuration

The configuration of Reset Pin cannot be changed and is not controllable through IOMUX. Its configuration is fixed as follows:

- Speed: 100 Mhz
- Slew Rate: Slow
- Open Drain: Disable (Output is CMOS)
- Hysteresis: Disable (CMOS input)
- Drive Strength: 25 Ohm
- Pull Up/Down: 100KOhm Pull-UP
- Pull/Keep Enable: Enable
- Pull/Keep Select: Pull Selected
- ibe/obe/do/ind: Controlled through Reset Controller

5.4.8 Typical IOMUX Configuration

The table below shows the typical IOMUX configuration of pads for different modules on this device. This is sample configuration only and will change based on the instance of the module and pads involved.

Table 5-239. Typical IOMUX Configuration

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
ENET	RMII0_CRS_DV	Input	0	0	0	0	0	7	0	0	1
	RMII0_MDIO	Input/Output	3	0	0	0	0	7	0	0	1
	RMII0_RXER	Input	0	0	0	0	0	7	0	0	1
	RMII0_RXD[0]	Input	0	0	0	0	0	7	0	0	1
	RMII0_RXD[1]	Input	0	0	0	0	0	7	0	0	1
	RMII0_MDC	Output	3	0	0	0	0	7	0	0	0
	RMII0_TXD[1]	Output	3	0	0	0	0	7	0	0	0
	RMII0_TXD[0]	Output	3	0	0	0	0	7	0	0	0
	RMII0_TXEN	Output	3	0	0	0	0	7	0	0	0
	RMII1_CRS_DV	Input	0	0	0	0	0	7	0	0	1
	RMII1_MDIO	Input/Output	3	0	0	0	0	7	0	0	1
	RMII1_RXER	Input	0	0	0	0	0	7	0	0	1
	RMII1_RXD[0]	Input	0	0	0	0	0	7	0	0	1
	RMII1_RXD[1]	Input	0	0	0	0	0	7	0	0	1
	RMII1_MDC	Output	3	0	0	0	0	7	0	0	0
	RMII1_TXD[1]	Output	3	0	0	0	0	7	0	0	0
	RMII1_TXD[0]	Output	3	0	0	0	0	7	0	0	0
	RMII1_TXEN	Output	3	0	0	0	0	7	0	0	0
	MAC0_TMR0	Input/Output	3	0	0	0	0	7	0	0	1
	MAC0_TMR1	Input/Output	3	0	0	0	0	7	0	0	1
	MAC0_TMR2	Input/Output	3	0	0	0	0	7	0	0	1
	MAC0_TMR3	Input/Output	3	0	0	0	0	7	0	0	1
	MAC1_T	Input/	3	0	0	0	0	7	0	0	1

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
SPDIF	IN1	Input	0	0	0	0	0	1	0	0	1
	OUT1	Output	3	0	0	0	0	7	0	0	0
	PLOCK	Output	3	0	0	0	0	7	0	0	0
	EXTCLK	Input	0	0	0	0	0	1	0	0	0
	SRCLK	Output	3	0	0	0	0	7	0	0	1
ESAI	FSR	Input/ Output	3	0	0	0	0	7	0	0	1
	FST	Input/ Output	3	0	0	0	0	7	0	0	1
	HCKR	Input/ Output	3	0	0	0	0	7	0	0	1
	HCKT	Input/ Output	3	0	0	0	0	7	0	0	1
	SCKR	Input/ Output	3	0	0	0	0	7	0	0	1
	SCKT	Input/ Output	3	0	0	0	0	7	0	0	1
	SDO0	Input/ Output	3	0	0	0	0	7	0	0	1
	SDO1	Input/ Output	3	0	0	0	0	7	0	0	1
	SDO2	Input/ Output	3	0	0	0	0	7	0	0	1
	SDO3	Input/ Output	3	0	0	0	0	7	0	0	1
	SDI1	Input/ Output	3	0	0	0	0	7	0	0	1
	SDI0	Input/ Output	3	0	0	0	0	7	0	0	1
SAI	RX_BCLK	Input/ Output	3	0	0	0	0	7	0	0	1
	TX_BCLK	Input/ Output	3	0	0	0	0	7	0	0	1
	RX_DATA	Input	0	0	0	0	0	1	0	0	1
	TX_DATA	Output	3	0	0	0	0	7	0	0	0
	RX_SYNC	Input/ Output	3	0	0	0	0	7	0	0	1
	TX_SYNC	Input/ Output	3	0	0	0	0	7	0	0	1

Table continues on the next page...

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
MLB	MLBDATA	Input/ Output	3	0	0	1	0	7	2	1	1
	MLBSIGNAL	Input/ Output	3	0	0	0	0	7	2	0	1
	MLBCLK	Input	0	0	0	1	0	7	2	1	1
FTM	CH[7:0]	Input/ Output	3	0	0	0	0	7	0	0	1
	QD_PHA	Input	3	0	0	0	0	7	0	0	1
	QD_PHB	Input	3	0	0	0	0	7	0	0	1
VIU	DATA[23:0]	Input	0	0	0	0	0	7	0	0	1
	PIX_CLK	Input	0	0	0	0	0	7	0	0	1
	VIU_FID	Input	0	0	0	0	0	7	0	0	1
	HSYNC	Input	0	0	0	0	0	7	0	0	1
	VSYNC	Input	0	0	0	0	0	7	0	0	1
	VIU_DE	Input	0	0	0	0	0	7	0	0	1
TCON	TCON[11:0]	Output	3	0	0	0	0	7	0	0	0
	DATA_OUT[1:0]	Output	3	0	0	0	0	7	0	0	0
	DATA_OUT[25:2]	Output	3	0	0	0	0	7	0	0	0
PDB	EXTRIG	Input	0	0	0	0	0	7	0	0	1
USB	VBUS_OC	Input	0	0	0	0	0	7	0	0	1
	VBUS_EN	Output	3	0	1	0	0	7	0	0	0
	VBUS_OC_OTG	Input	0	0	0	0	0	7	0	0	1
	VBUS_EN_OTG	Output	3	0	1	0	0	7	0	0	0
	USB0_SOF_PULSE	Output	3	0	1	0	0	7	0	0	0
	USB1_SOF_PULSE	Output	3	0	1	0	0	7	0	0	0
SNVS	SNVS_ALARM_OUT_B	Output	3	0	0	0	0	7	2	0	0

Table continues on the next page...

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
LCD	LCD0	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD1	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD2	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD3	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD4	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD5	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD6	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD7	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD8	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD9	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD10	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD11	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD12	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD13	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD14	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD15	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD16	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD17	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD18	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD19	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD20	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD21	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD22	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD23	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD24	Input/ Output	0	0	0	0	0	1	0	0	0
	LCD25	Input/	0	0	0	0	0	1	0	0	0

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
ADC0	ADC0SE0	Input/ Output	0	0	0	0	0	1	0	0	0
	ADC0SE1	Input/ Output	0	0	0	0	0	1	0	0	0
	ADC0SE2	Input/ Output	0	0	0	0	0	1	0	0	0
	ADC0SE3	Input/ Output	0	0	0	0	0	1	0	0	0
	ADC0SE4	Input/ Output	0	0	0	0	0	1	0	0	0
	ADC0SE5	Input/ Output	0	0	0	0	0	1	0	0	0
	ADC0SE6	Input/ Output	0	0	0	0	0	1	0	0	0
	ADC0SE7	Input/ Output	0	0	0	0	0	1	0	0	0

Table continues on the next page...

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
DDR	DDR_RESETB	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_A_15:0	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_BA_2:0	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_CAS_B	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_CKE_0	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_CLK_0	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_CS_B_0	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_D_15:0	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_DQM_1:0	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_DQS_1:0	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_RAS_B	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_WE_B	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DDR_ODT_1:0	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DUMMY_DDRBYTE1	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA
	DUMMY_DDRBYTE2	DDR Pad - No IBE/OBE	0	0	0	0	0	7	0	0	NA

Table continues on the next page...

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
SCI	TX	Output	3	0	0	1	0	7	2	1	0
	RX	Input	3	0	0	1	0	7	2	1	1
	RTS	Output	3	0	0	1	0	7	2	1	0
	CTS	Input	3	0	0	1	0	7	2	1	1
I2C	SCL	Input/ Output	3	1	1	1	0	7	2	0	1
	SDA	Input/ Output	3	1	1	1	0	7	2	0	1
SDHC	DAT0	Input/ Output	3	0	0	1	0	7	2	1	1
	DAT1	Input/ Output	3	0	0	1	0	7	2	1	1
	DAT2	Input/ Output	3	0	0	1	0	7	2	1	1
	DAT3	Input/ Output	3	0	0	1	0	7	2	1	1
	DAT4	Input/ Output	3	0	0	1	0	7	2	1	1
	DAT5	Input/ Output	3	0	0	1	0	7	2	1	1
	DAT6	Input/ Output	3	0	0	1	0	7	2	1	1
	DAT7	Input/ Output	3	0	0	1	0	7	2	1	1
	CMD	Input/ Output	3	0	0	1	0	7	2	1	1
	CLK	Output	3	0	0	1	0	7	2	1	1
	WP	Input	3	0	0	1	0	7	0	1	1
SPI	PCS3	Output	3	1	0	0	0	7	2	0	0
	PCS2	Output	3	1	0	0	0	7	2	0	0
	PCS1	Output	3	1	0	0	0	7	2	0	0
	PCS0	Output	3	1	0	0	0	7	2	0	0
	SIN	Input	3	1	0	0	0	7	2	0	1
	SOUT	Output	3	1	0	0	0	7	2	0	0
	SCK	Output	3	1	0	0	0	7	2	0	0

Table continues on the next page...

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
FlexBus	FB_AD[31]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[30]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[29]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[28]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[27]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[26]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[25]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[24]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[23]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[22]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[21]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[20]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[19]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[18]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[17]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[16]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[15]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[14]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[13]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[12]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[11]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[10]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[9]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[8]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[7]	Input/ Output	3	0	0	0	0	7	2	0	1
	FB_AD[6	Input/	3	0	0	0	0	7	2	0	1

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
	FB_BE2_b	Output	3	0	0	0	0	7	2	0	0
	FB_BE1_b	Output	3	0	0	0	0	7	2	0	0
	FB_BE0_b	Output	3	0	0	0	0	7	2	0	0
QSPI	A_SCK	Output	3	0	0	0	0	7	2	0	0
	A_CS0	Output	3	0	0	0	0	7	2	0	0
	A_CS1	Output	3	0	0	0	0	7	2	0	0
	A_DATA[3]	Input/Output	3	0	0	0	0	7	2	0	1
	A_DATA[2]	Input/Output	3	0	0	0	0	7	2	0	1
	A_DATA[1]	Input/Output	3	0	0	0	0	7	2	0	1
	A_DATA[0]	Input/Output	3	0	0	0	0	7	2	0	1
	A_DQS	Output	3	0	0	0	0	7	2	0	0
	B_SCK	Output	3	0	0	0	0	7	2	0	0
	B_CS0	Output	3	0	0	0	0	7	2	0	0
	B_CS1	Output	3	0	0	0	0	7	2	0	0
	B_DATA[3]	Input/Output	3	0	0	0	0	7	2	0	1
	B_DATA[2]	Input/Output	3	0	0	0	0	7	2	0	1
	B_DATA[1]	Input/Output	3	0	0	0	0	7	2	0	1
	B_DATA[0]	Input/Output	3	0	0	0	0	7	2	0	1
	B_DQS	Output	3	0	0	0	0	7	2	0	0

Table continues on the next page...

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
NFC	NF_ALE	Output	3	0	0	0	0	7	2	0	0
	NF_CLE	Output	3	0	0	0	0	7	2	0	0
	NF_CE0_b	Output	3	0	0	0	0	7	2	0	0
	NF_CE1_b	Output	3	0	0	0	0	7	2	0	0
	NF_IO[15]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[14]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[13]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[12]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[11]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[10]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[9]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[8]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[7]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[6]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[5]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[4]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[3]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[2]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[1]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_IO[0]	Input/ Output	3	0	0	0	0	7	2	0	1
	NF_R/B_b	Input	3	0	0	0	0	7	2	0	1
	NF_RE_b	Output	3	0	0	0	0	7	2	0	0
	NF_WE_b	Output	3	0	0	0	0	7	2	0	0

Table continues on the next page...

Table 5-239. Typical IOMUX Configuration (continued)

Module Name	Signal Name	Direction	SPEED	HYS	ODE	PUE	SRE	DSE	PUS	PKE	IBE
FlexCAN	RX	Input	3	0	0	0	0	7	2	0	1
	TX	Output	3	0	0	0	0	7	2	0	0

Chapter 6
Port control and interrupts (PORT)

6.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

6.2 Overview

The port control and interrupt (PORT) module provides support for digital filtering, and external interrupt functions. Most functions can be configured independently for each pin in the 32-bit port and affect the pin regardless of its pin muxing state.

There is one instance of the PORT module for each port. Not all pins within each port are implemented on a specific device.

6.2.1 Features

The PORT module has the following features:

- Pin interrupt
  - Interrupt flag and enable registers for each pin
  - Support for edge sensitive (rising, falling, both) or level sensitive (low, high) configured per pin
  - Support for interrupt or DMA request configured per pin
  - Asynchronous wakeup in Low-Power modes
  - Pin interrupt is functional in all digital Pin Muxing modes
- Digital input filter

- Digital input filter for each pin, usable by any digital peripheral muxed onto the pin
- Individual enable or bypass control field per pin
- Selectable clock source for digital input filter with a five bit resolution on filter size
- Functional in all digital Pin Muxing modes

6.2.2 Modes of operation

6.2.2.1 Run mode

In Run mode, the PORT operates normally.

6.2.2.2 Wait mode

In Wait mode, PORT continues to operate normally and may be configured to exit the Low-Power mode if an enabled interrupt is detected. DMA requests are still generated during the Wait mode, but do not cause an exit from the Low-Power mode.

6.2.2.3 Stop mode

In Stop mode, the PORT can be configured to exit the Low-Power mode via an asynchronous wakeup signal if an enabled interrupt is detected.

In Stop mode, the digital input filters are bypassed unless they are configured to run from the 1 kHz LPO clock source.

6.2.2.4 Debug mode

In Debug mode, PORT operates normally.

6.3 External signal description

The following table describes the PORT external signal.

Table 6-1. Signal properties

Name	Function	I/O	Reset	Pull
PORTx[31:0]	External interrupt	I/O	0	-

NOTE

Not all pins within each port are implemented on each device.

6.4 Detailed signal description

The following table contains the detailed signal description for the PORT interface.

Table 6-2. PORT interface—detailed signal description

Signal	I/O	Description	
PORTx[31:0]	I/O	External interrupt.	
		State meaning	Asserted—pin is logic one. Negated—pin is logic zero.
		Timing	Assertion—may occur at any time and can assert asynchronously to the system clock. Negation—may occur at any time and can assert asynchronously to the system clock.

6.5 Memory map and register definition

Any read or write access to the PORT memory space that is outside the valid memory map results in a bus error. All register accesses complete with zero wait states.

PORT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_9000	Pin Control Register n (PORT0_PCR0)	32	R/W	0000_0000h	6.5.1/525
4004_9004	Pin Control Register n (PORT0_PCR1)	32	R/W	0000_0000h	6.5.1/525
4004_9008	Pin Control Register n (PORT0_PCR2)	32	R/W	0000_0000h	6.5.1/525
4004_900C	Pin Control Register n (PORT0_PCR3)	32	R/W	0000_0000h	6.5.1/525
4004_9010	Pin Control Register n (PORT0_PCR4)	32	R/W	0000_0000h	6.5.1/525
4004_9014	Pin Control Register n (PORT0_PCR5)	32	R/W	0000_0000h	6.5.1/525
4004_9018	Pin Control Register n (PORT0_PCR6)	32	R/W	0000_0000h	6.5.1/525
4004_901C	Pin Control Register n (PORT0_PCR7)	32	R/W	0000_0000h	6.5.1/525

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_9020	Pin Control Register n (PORT0_PCR8)	32	R/W	0000_0000h	6.5.1/525
4004_9024	Pin Control Register n (PORT0_PCR9)	32	R/W	0000_0000h	6.5.1/525
4004_9028	Pin Control Register n (PORT0_PCR10)	32	R/W	0000_0000h	6.5.1/525
4004_902C	Pin Control Register n (PORT0_PCR11)	32	R/W	0000_0000h	6.5.1/525
4004_9030	Pin Control Register n (PORT0_PCR12)	32	R/W	0000_0000h	6.5.1/525
4004_9034	Pin Control Register n (PORT0_PCR13)	32	R/W	0000_0000h	6.5.1/525
4004_9038	Pin Control Register n (PORT0_PCR14)	32	R/W	0000_0000h	6.5.1/525
4004_903C	Pin Control Register n (PORT0_PCR15)	32	R/W	0000_0000h	6.5.1/525
4004_9040	Pin Control Register n (PORT0_PCR16)	32	R/W	0000_0000h	6.5.1/525
4004_9044	Pin Control Register n (PORT0_PCR17)	32	R/W	0000_0000h	6.5.1/525
4004_9048	Pin Control Register n (PORT0_PCR18)	32	R/W	0000_0000h	6.5.1/525
4004_904C	Pin Control Register n (PORT0_PCR19)	32	R/W	0000_0000h	6.5.1/525
4004_9050	Pin Control Register n (PORT0_PCR20)	32	R/W	0000_0000h	6.5.1/525
4004_9054	Pin Control Register n (PORT0_PCR21)	32	R/W	0000_0000h	6.5.1/525
4004_9058	Pin Control Register n (PORT0_PCR22)	32	R/W	0000_0000h	6.5.1/525
4004_905C	Pin Control Register n (PORT0_PCR23)	32	R/W	0000_0000h	6.5.1/525
4004_9060	Pin Control Register n (PORT0_PCR24)	32	R/W	0000_0000h	6.5.1/525
4004_9064	Pin Control Register n (PORT0_PCR25)	32	R/W	0000_0000h	6.5.1/525
4004_9068	Pin Control Register n (PORT0_PCR26)	32	R/W	0000_0000h	6.5.1/525
4004_906C	Pin Control Register n (PORT0_PCR27)	32	R/W	0000_0000h	6.5.1/525
4004_9070	Pin Control Register n (PORT0_PCR28)	32	R/W	0000_0000h	6.5.1/525
4004_9074	Pin Control Register n (PORT0_PCR29)	32	R/W	0000_0000h	6.5.1/525
4004_9078	Pin Control Register n (PORT0_PCR30)	32	R/W	0000_0000h	6.5.1/525
4004_907C	Pin Control Register n (PORT0_PCR31)	32	R/W	0000_0000h	6.5.1/525
4004_90A0	Interrupt Status Flag Register (PORT0_ISFR)	32	w1c	0000_0000h	6.5.2/526
4004_90C0	Digital Filter Enable Register (PORT0_DFER)	32	R/W	0000_0000h	6.5.3/527
4004_90C4	Digital Filter Clock Register (PORT0_DFCR)	32	R/W	0000_0000h	6.5.4/527
4004_90C8	Digital Filter Width Register (PORT0_DFWR)	32	R/W	0000_0000h	6.5.5/528
4004_A000	Pin Control Register n (PORT1_PCR0)	32	R/W	0000_0000h	6.5.1/525
4004_A004	Pin Control Register n (PORT1_PCR1)	32	R/W	0000_0000h	6.5.1/525
4004_A008	Pin Control Register n (PORT1_PCR2)	32	R/W	0000_0000h	6.5.1/525
4004_A00C	Pin Control Register n (PORT1_PCR3)	32	R/W	0000_0000h	6.5.1/525
4004_A010	Pin Control Register n (PORT1_PCR4)	32	R/W	0000_0000h	6.5.1/525
4004_A014	Pin Control Register n (PORT1_PCR5)	32	R/W	0000_0000h	6.5.1/525
4004_A018	Pin Control Register n (PORT1_PCR6)	32	R/W	0000_0000h	6.5.1/525
4004_A01C	Pin Control Register n (PORT1_PCR7)	32	R/W	0000_0000h	6.5.1/525
4004_A020	Pin Control Register n (PORT1_PCR8)	32	R/W	0000_0000h	6.5.1/525
4004_A024	Pin Control Register n (PORT1_PCR9)	32	R/W	0000_0000h	6.5.1/525

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_A028	Pin Control Register n (PORT1_PCR10)	32	R/W	0000_0000h	6.5.1/525
4004_A02C	Pin Control Register n (PORT1_PCR11)	32	R/W	0000_0000h	6.5.1/525
4004_A030	Pin Control Register n (PORT1_PCR12)	32	R/W	0000_0000h	6.5.1/525
4004_A034	Pin Control Register n (PORT1_PCR13)	32	R/W	0000_0000h	6.5.1/525
4004_A038	Pin Control Register n (PORT1_PCR14)	32	R/W	0000_0000h	6.5.1/525
4004_A03C	Pin Control Register n (PORT1_PCR15)	32	R/W	0000_0000h	6.5.1/525
4004_A040	Pin Control Register n (PORT1_PCR16)	32	R/W	0000_0000h	6.5.1/525
4004_A044	Pin Control Register n (PORT1_PCR17)	32	R/W	0000_0000h	6.5.1/525
4004_A048	Pin Control Register n (PORT1_PCR18)	32	R/W	0000_0000h	6.5.1/525
4004_A04C	Pin Control Register n (PORT1_PCR19)	32	R/W	0000_0000h	6.5.1/525
4004_A050	Pin Control Register n (PORT1_PCR20)	32	R/W	0000_0000h	6.5.1/525
4004_A054	Pin Control Register n (PORT1_PCR21)	32	R/W	0000_0000h	6.5.1/525
4004_A058	Pin Control Register n (PORT1_PCR22)	32	R/W	0000_0000h	6.5.1/525
4004_A05C	Pin Control Register n (PORT1_PCR23)	32	R/W	0000_0000h	6.5.1/525
4004_A060	Pin Control Register n (PORT1_PCR24)	32	R/W	0000_0000h	6.5.1/525
4004_A064	Pin Control Register n (PORT1_PCR25)	32	R/W	0000_0000h	6.5.1/525
4004_A068	Pin Control Register n (PORT1_PCR26)	32	R/W	0000_0000h	6.5.1/525
4004_A06C	Pin Control Register n (PORT1_PCR27)	32	R/W	0000_0000h	6.5.1/525
4004_A070	Pin Control Register n (PORT1_PCR28)	32	R/W	0000_0000h	6.5.1/525
4004_A074	Pin Control Register n (PORT1_PCR29)	32	R/W	0000_0000h	6.5.1/525
4004_A078	Pin Control Register n (PORT1_PCR30)	32	R/W	0000_0000h	6.5.1/525
4004_A07C	Pin Control Register n (PORT1_PCR31)	32	R/W	0000_0000h	6.5.1/525
4004_A0A0	Interrupt Status Flag Register (PORT1_ISFR)	32	w1c	0000_0000h	6.5.2/526
4004_A0C0	Digital Filter Enable Register (PORT1_DFER)	32	R/W	0000_0000h	6.5.3/527
4004_A0C4	Digital Filter Clock Register (PORT1_DFCR)	32	R/W	0000_0000h	6.5.4/527
4004_A0C8	Digital Filter Width Register (PORT1_DFWR)	32	R/W	0000_0000h	6.5.5/528
4004_B000	Pin Control Register n (PORT2_PCR0)	32	R/W	0000_0000h	6.5.1/525
4004_B004	Pin Control Register n (PORT2_PCR1)	32	R/W	0000_0000h	6.5.1/525
4004_B008	Pin Control Register n (PORT2_PCR2)	32	R/W	0000_0000h	6.5.1/525
4004_B00C	Pin Control Register n (PORT2_PCR3)	32	R/W	0000_0000h	6.5.1/525
4004_B010	Pin Control Register n (PORT2_PCR4)	32	R/W	0000_0000h	6.5.1/525
4004_B014	Pin Control Register n (PORT2_PCR5)	32	R/W	0000_0000h	6.5.1/525
4004_B018	Pin Control Register n (PORT2_PCR6)	32	R/W	0000_0000h	6.5.1/525
4004_B01C	Pin Control Register n (PORT2_PCR7)	32	R/W	0000_0000h	6.5.1/525
4004_B020	Pin Control Register n (PORT2_PCR8)	32	R/W	0000_0000h	6.5.1/525
4004_B024	Pin Control Register n (PORT2_PCR9)	32	R/W	0000_0000h	6.5.1/525
4004_B028	Pin Control Register n (PORT2_PCR10)	32	R/W	0000_0000h	6.5.1/525
4004_B02C	Pin Control Register n (PORT2_PCR11)	32	R/W	0000_0000h	6.5.1/525

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_B030	Pin Control Register n (PORT2_PCR12)	32	R/W	0000_0000h	6.5.1/525
4004_B034	Pin Control Register n (PORT2_PCR13)	32	R/W	0000_0000h	6.5.1/525
4004_B038	Pin Control Register n (PORT2_PCR14)	32	R/W	0000_0000h	6.5.1/525
4004_B03C	Pin Control Register n (PORT2_PCR15)	32	R/W	0000_0000h	6.5.1/525
4004_B040	Pin Control Register n (PORT2_PCR16)	32	R/W	0000_0000h	6.5.1/525
4004_B044	Pin Control Register n (PORT2_PCR17)	32	R/W	0000_0000h	6.5.1/525
4004_B048	Pin Control Register n (PORT2_PCR18)	32	R/W	0000_0000h	6.5.1/525
4004_B04C	Pin Control Register n (PORT2_PCR19)	32	R/W	0000_0000h	6.5.1/525
4004_B050	Pin Control Register n (PORT2_PCR20)	32	R/W	0000_0000h	6.5.1/525
4004_B054	Pin Control Register n (PORT2_PCR21)	32	R/W	0000_0000h	6.5.1/525
4004_B058	Pin Control Register n (PORT2_PCR22)	32	R/W	0000_0000h	6.5.1/525
4004_B05C	Pin Control Register n (PORT2_PCR23)	32	R/W	0000_0000h	6.5.1/525
4004_B060	Pin Control Register n (PORT2_PCR24)	32	R/W	0000_0000h	6.5.1/525
4004_B064	Pin Control Register n (PORT2_PCR25)	32	R/W	0000_0000h	6.5.1/525
4004_B068	Pin Control Register n (PORT2_PCR26)	32	R/W	0000_0000h	6.5.1/525
4004_B06C	Pin Control Register n (PORT2_PCR27)	32	R/W	0000_0000h	6.5.1/525
4004_B070	Pin Control Register n (PORT2_PCR28)	32	R/W	0000_0000h	6.5.1/525
4004_B074	Pin Control Register n (PORT2_PCR29)	32	R/W	0000_0000h	6.5.1/525
4004_B078	Pin Control Register n (PORT2_PCR30)	32	R/W	0000_0000h	6.5.1/525
4004_B07C	Pin Control Register n (PORT2_PCR31)	32	R/W	0000_0000h	6.5.1/525
4004_B0A0	Interrupt Status Flag Register (PORT2_ISFR)	32	w1c	0000_0000h	6.5.2/526
4004_B0C0	Digital Filter Enable Register (PORT2_DFER)	32	R/W	0000_0000h	6.5.3/527
4004_B0C4	Digital Filter Clock Register (PORT2_DFCR)	32	R/W	0000_0000h	6.5.4/527
4004_B0C8	Digital Filter Width Register (PORT2_DFWR)	32	R/W	0000_0000h	6.5.5/528
4004_C000	Pin Control Register n (PORT3_PCR0)	32	R/W	0000_0000h	6.5.1/525
4004_C004	Pin Control Register n (PORT3_PCR1)	32	R/W	0000_0000h	6.5.1/525
4004_C008	Pin Control Register n (PORT3_PCR2)	32	R/W	0000_0000h	6.5.1/525
4004_C00C	Pin Control Register n (PORT3_PCR3)	32	R/W	0000_0000h	6.5.1/525
4004_C010	Pin Control Register n (PORT3_PCR4)	32	R/W	0000_0000h	6.5.1/525
4004_C014	Pin Control Register n (PORT3_PCR5)	32	R/W	0000_0000h	6.5.1/525
4004_C018	Pin Control Register n (PORT3_PCR6)	32	R/W	0000_0000h	6.5.1/525
4004_C01C	Pin Control Register n (PORT3_PCR7)	32	R/W	0000_0000h	6.5.1/525
4004_C020	Pin Control Register n (PORT3_PCR8)	32	R/W	0000_0000h	6.5.1/525
4004_C024	Pin Control Register n (PORT3_PCR9)	32	R/W	0000_0000h	6.5.1/525
4004_C028	Pin Control Register n (PORT3_PCR10)	32	R/W	0000_0000h	6.5.1/525
4004_C02C	Pin Control Register n (PORT3_PCR11)	32	R/W	0000_0000h	6.5.1/525
4004_C030	Pin Control Register n (PORT3_PCR12)	32	R/W	0000_0000h	6.5.1/525
4004_C034	Pin Control Register n (PORT3_PCR13)	32	R/W	0000_0000h	6.5.1/525

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_C038	Pin Control Register n (PORT3_PCR14)	32	R/W	0000_0000h	6.5.1/525
4004_C03C	Pin Control Register n (PORT3_PCR15)	32	R/W	0000_0000h	6.5.1/525
4004_C040	Pin Control Register n (PORT3_PCR16)	32	R/W	0000_0000h	6.5.1/525
4004_C044	Pin Control Register n (PORT3_PCR17)	32	R/W	0000_0000h	6.5.1/525
4004_C048	Pin Control Register n (PORT3_PCR18)	32	R/W	0000_0000h	6.5.1/525
4004_C04C	Pin Control Register n (PORT3_PCR19)	32	R/W	0000_0000h	6.5.1/525
4004_C050	Pin Control Register n (PORT3_PCR20)	32	R/W	0000_0000h	6.5.1/525
4004_C054	Pin Control Register n (PORT3_PCR21)	32	R/W	0000_0000h	6.5.1/525
4004_C058	Pin Control Register n (PORT3_PCR22)	32	R/W	0000_0000h	6.5.1/525
4004_C05C	Pin Control Register n (PORT3_PCR23)	32	R/W	0000_0000h	6.5.1/525
4004_C060	Pin Control Register n (PORT3_PCR24)	32	R/W	0000_0000h	6.5.1/525
4004_C064	Pin Control Register n (PORT3_PCR25)	32	R/W	0000_0000h	6.5.1/525
4004_C068	Pin Control Register n (PORT3_PCR26)	32	R/W	0000_0000h	6.5.1/525
4004_C06C	Pin Control Register n (PORT3_PCR27)	32	R/W	0000_0000h	6.5.1/525
4004_C070	Pin Control Register n (PORT3_PCR28)	32	R/W	0000_0000h	6.5.1/525
4004_C074	Pin Control Register n (PORT3_PCR29)	32	R/W	0000_0000h	6.5.1/525
4004_C078	Pin Control Register n (PORT3_PCR30)	32	R/W	0000_0000h	6.5.1/525
4004_C07C	Pin Control Register n (PORT3_PCR31)	32	R/W	0000_0000h	6.5.1/525
4004_C0A0	Interrupt Status Flag Register (PORT3_ISFR)	32	w1c	0000_0000h	6.5.2/526
4004_C0C0	Digital Filter Enable Register (PORT3_DFER)	32	R/W	0000_0000h	6.5.3/527
4004_C0C4	Digital Filter Clock Register (PORT3_DFCR)	32	R/W	0000_0000h	6.5.4/527
4004_C0C8	Digital Filter Width Register (PORT3_DFWR)	32	R/W	0000_0000h	6.5.5/528
4004_D000	Pin Control Register n (PORT4_PCR0)	32	R/W	0000_0000h	6.5.1/525
4004_D004	Pin Control Register n (PORT4_PCR1)	32	R/W	0000_0000h	6.5.1/525
4004_D008	Pin Control Register n (PORT4_PCR2)	32	R/W	0000_0000h	6.5.1/525
4004_D00C	Pin Control Register n (PORT4_PCR3)	32	R/W	0000_0000h	6.5.1/525
4004_D010	Pin Control Register n (PORT4_PCR4)	32	R/W	0000_0000h	6.5.1/525
4004_D014	Pin Control Register n (PORT4_PCR5)	32	R/W	0000_0000h	6.5.1/525
4004_D018	Pin Control Register n (PORT4_PCR6)	32	R/W	0000_0000h	6.5.1/525
4004_D01C	Pin Control Register n (PORT4_PCR7)	32	R/W	0000_0000h	6.5.1/525
4004_D020	Pin Control Register n (PORT4_PCR8)	32	R/W	0000_0000h	6.5.1/525
4004_D024	Pin Control Register n (PORT4_PCR9)	32	R/W	0000_0000h	6.5.1/525
4004_D028	Pin Control Register n (PORT4_PCR10)	32	R/W	0000_0000h	6.5.1/525
4004_D02C	Pin Control Register n (PORT4_PCR11)	32	R/W	0000_0000h	6.5.1/525
4004_D030	Pin Control Register n (PORT4_PCR12)	32	R/W	0000_0000h	6.5.1/525
4004_D034	Pin Control Register n (PORT4_PCR13)	32	R/W	0000_0000h	6.5.1/525
4004_D038	Pin Control Register n (PORT4_PCR14)	32	R/W	0000_0000h	6.5.1/525
4004_D03C	Pin Control Register n (PORT4_PCR15)	32	R/W	0000_0000h	6.5.1/525

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_D040	Pin Control Register n (PORT4_PCR16)	32	R/W	0000_0000h	6.5.1/525
4004_D044	Pin Control Register n (PORT4_PCR17)	32	R/W	0000_0000h	6.5.1/525
4004_D048	Pin Control Register n (PORT4_PCR18)	32	R/W	0000_0000h	6.5.1/525
4004_D04C	Pin Control Register n (PORT4_PCR19)	32	R/W	0000_0000h	6.5.1/525
4004_D050	Pin Control Register n (PORT4_PCR20)	32	R/W	0000_0000h	6.5.1/525
4004_D054	Pin Control Register n (PORT4_PCR21)	32	R/W	0000_0000h	6.5.1/525
4004_D058	Pin Control Register n (PORT4_PCR22)	32	R/W	0000_0000h	6.5.1/525
4004_D05C	Pin Control Register n (PORT4_PCR23)	32	R/W	0000_0000h	6.5.1/525
4004_D060	Pin Control Register n (PORT4_PCR24)	32	R/W	0000_0000h	6.5.1/525
4004_D064	Pin Control Register n (PORT4_PCR25)	32	R/W	0000_0000h	6.5.1/525
4004_D068	Pin Control Register n (PORT4_PCR26)	32	R/W	0000_0000h	6.5.1/525
4004_D06C	Pin Control Register n (PORT4_PCR27)	32	R/W	0000_0000h	6.5.1/525
4004_D070	Pin Control Register n (PORT4_PCR28)	32	R/W	0000_0000h	6.5.1/525
4004_D074	Pin Control Register n (PORT4_PCR29)	32	R/W	0000_0000h	6.5.1/525
4004_D078	Pin Control Register n (PORT4_PCR30)	32	R/W	0000_0000h	6.5.1/525
4004_D07C	Pin Control Register n (PORT4_PCR31)	32	R/W	0000_0000h	6.5.1/525
4004_D0A0	Interrupt Status Flag Register (PORT4_ISFR)	32	w1c	0000_0000h	6.5.2/526
4004_D0C0	Digital Filter Enable Register (PORT4_DFER)	32	R/W	0000_0000h	6.5.3/527
4004_D0C4	Digital Filter Clock Register (PORT4_DFCR)	32	R/W	0000_0000h	6.5.4/527
4004_D0C8	Digital Filter Width Register (PORT4_DFWR)	32	R/W	0000_0000h	6.5.5/528

6.5.1 Pin Control Register n (PORTx_PCRn)

NOTE

Refer to the Signal Multiplexing and Pin Assignment chapter for the reset value of this device.

See the GPIO Configuration section for details on the available functions for each pin.

Do not modify pin configuration registers associated with pins not available in your selected package. All un-bonded pins not available in your package will default to DISABLE state for lowest power consumption.

Address: Base address + 0h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ISF	0				IRQC			
W								w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_PCRn field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0 Configured interrupt is not detected. 1 Configured interrupt is detected. If the pin is configured to generate a DMA request, then the corresponding flag will be cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic one is written to the flag. If the pin is configured for a level sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 IRQC	Interrupt Configuration

Table continues on the next page...

PORTx_PCR*n* field descriptions (continued)

Field	Description
	The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt/DMA request as follows: 0000 Interrupt/DMA request disabled. 0001 DMA request on rising edge. 0010 DMA request on falling edge. 0011 DMA request on either edge. 1000 Interrupt when logic zero. 1001 Interrupt on rising edge. 1010 Interrupt on falling edge. 1011 Interrupt on either edge. 1100 Interrupt when logic one. Others Reserved.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

6.5.2 Interrupt Status Flag Register (PORTx_ISFR)

The pin interrupt configuration is valid in all digital pin muxing modes. The Interrupt Status Flag for each pin is also visible in the corresponding Pin Control Register, and each flag can be cleared in either location.

Address: Base address + A0h offset

Bit	31–0
R	ISF
W	w1c
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PORTx_ISFR field descriptions

Field	Description
31–0 ISF	Interrupt Status Flag Each bit in the field indicates the detection of the configured interrupt of the same number as the field. 0 Configured interrupt is not detected. 1 Configured interrupt is detected. If the pin is configured to generate a DMA request, then the corresponding flag will be cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic one is written to the flag. If the pin is configured for a level sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared.

6.5.3 Digital Filter Enable Register (PORTx_DFER)

The digital filter configuration is valid in all digital pin muxing modes.

Address: Base address + C0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	DFE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_DFER field descriptions

Field	Description
31–0 DFE	Digital Filter Enable The digital filter configuration is valid in all digital pin muxing modes. The output of each digital filter is reset to zero at system reset and whenever the digital filter is disabled. Each bit in the field enables the digital filter of the same number as the field. 0 Digital filter is disabled on the corresponding pin and output of the digital filter is reset to zero. 1 Digital filter is enabled on the corresponding pin, if the pin is configured as a digital input.

6.5.4 Digital Filter Clock Register (PORTx_DFCR)

The digital filter configuration is valid in all digital pin muxing modes.

Address: Base address + C4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															CS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_DFCR field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 CS	Clock Source

Table continues on the next page...

PORTx_DFCR field descriptions (continued)

Field	Description
	The digital filter configuration is valid in all digital pin muxing modes. Configures the clock source for the digital input filters. Changing the filter clock source must be done only when all digital filters are disabled. 0 Digital filters are clocked by the bus clock. 1 Digital filters are clocked by the 1 kHz LPO clock.

6.5.5 Digital Filter Width Register (PORTx_DFWR)

The digital filter configuration is valid in all digital pin muxing modes.

Address: Base address + C8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											FILT				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_DFWR field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 FILT	Filter Length The digital filter configuration is valid in all digital pin muxing modes. Configures the maximum size of the glitches, in clock cycles, that the digital filter absorbs for the enabled digital filters. Glitches that are longer than this register setting will pass through the digital filter, and glitches that are equal to or less than this register setting are filtered. Changing the filter length must be done only after all filters are disabled.

6.6 Functional description

6.6.1 External interrupts

The external interrupt capability of the PORT module is available in all digital pin muxing modes provided the PORT module is enabled.

Each pin can be individually configured for any of the following external interrupt modes:

- Interrupt disabled, default out of reset
- Active high level sensitive interrupt

- Active low level sensitive interrupt
- Rising edge sensitive interrupt
- Falling edge sensitive interrupt
- Rising and falling edge sensitive interrupt
- Rising edge sensitive DMA request
- Falling edge sensitive DMA request
- Rising and falling edge sensitive DMA request

The interrupt status flag is set when the configured edge or level is detected on the pin or at the output of the digital input filter, if the digital input digital filter is enabled. When not in Stop mode, the input is first synchronized to the bus clock to detect the configured level or edge transition.

The PORT module generates a single interrupt that asserts when the interrupt status flag is set for any enabled interrupt for that port. The interrupt negates after the interrupt status flags for all enabled interrupts have been cleared by writing a logic 1 to the ISF flag in either the PORT_ISFR or PORT_PCRn registers.

The PORT module generates a single DMA request that asserts when the interrupt status flag is set for any enabled DMA request in that port. The DMA request negates after the DMA transfer is completed, because that clears the interrupt status flags for all enabled DMA requests.

During Stop mode, the interrupt status flag for any enabled interrupt is asynchronously set if the required level or edge is detected. This also generates an asynchronous wakeup signal to exit the Low-Power mode.

6.6.2 Digital filter

The digital filter capabilities of the PORT module are available in all digital Pin Muxing modes if the PORT module is enabled.

The clock used for all digital filters within one port can be configured between the bus clock or the 1 kHz LPO clock. This selection must be changed only when all digital filters for that port are disabled. If the digital filters for a port are configured to use the bus clock, then the digital filters are bypassed for the duration of Stop mode. While the digital filters are bypassed, the output of each digital filter always equals the input pin, but the internal state of the digital filters remains static and does not update due to any change on the input pin.

The filter width in clock size is the same for all enabled digital filters within one port and must be changed only when all digital filters for that port are disabled.

The output of each digital filter is logic zero after system reset and whenever a digital filter is disabled. After a digital filter is enabled, the input is synchronized to the filter clock, either the bus clock or the 1 kHz LPO clock. If the synchronized input and the output of the digital filter remain different for a number of filter clock cycles equal to the filter width register configuration, then the output of the digital filter updates to equal the synchronized filter input.

The minimum latency through a digital filter equals two or three filter clock cycles plus the filter width configuration register.

Chapter 7
General-Purpose Input/Output (GPIO)

7.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The general-purpose input and output (GPIO) module communicates to the processor core via a zero wait state interface (IOPORT) for maximum pin performance. The GPIO registers support 8-bit, 16-bit or 32-bit accesses.

The GPIO data direction and output data registers control the direction and output data of each pin when the pin is configured for the GPIO function. The GPIO input data register displays the logic value on each pin when the pin is configured for any digital function, provided the corresponding Port Control and Interrupt module for that pin is enabled.

Efficient bit manipulation of the general-purpose outputs is supported through the addition of set, clear, and toggle write-only registers for each port output data register.

7.1.1 Features

- Features of the GPIO module include:
  - Pin input data register visible in all digital pin-multiplexing modes
  - Pin output data register with corresponding set/clear/toggle registers
  - Zero wait state access to GPIO registers through IOPORT

NOTE

GPIO module is clocked by system clock.

7.1.2 Modes of operation

The following table depicts different modes of operation and the behavior of the GPIO module in these modes.

Table 7-1. Modes of operation

Modes of operation	Description
Run	The GPIO module operates normally.
Wait	The GPIO module operates normally.
Stop	The GPIO module is disabled.
Debug	The GPIO module operates normally.

7.1.3 GPIO signal descriptions

Table 7-2. GPIO signal descriptions

GPIO signal descriptions	Description	I/O
PORT0[31:0]	General-purpose input/output	I/O
PORT1[31:0]	General-purpose input/output	I/O
PORT2[31:0]	General-purpose input/output	I/O
PORT3[31:0]	General-purpose input/output	I/O
PORT4[31:0]	General-purpose input/output	I/O

NOTE

Not all pins within each port are implemented on each device. See the chapter on signal multiplexing for the number of GPIO ports available in the device.

7.1.3.1 Detailed signal description

Table 7-3. GPIO interface-detailed signal descriptions

Signal	I/O	Description	
PORT0[31:0] PORT1[31:0] PORT2[31:0] PORT3[31:0] PORT4[31:0]	I/O	General-purpose input/output	
		State meaning	Asserted: The pin is logic 1. Deasserted: The pin is logic 0.
		Timing	Assertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock. Deassertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock.

7.2 Memory map and register definition

Any read or write access to the GPIO memory space that is outside the valid memory map results in a bus error.

NOTE

For simplicity, each GPIO port's registers appear with the same width of 32 bits, corresponding to 32 pins. The actual number of pins per port (and therefore the number of usable control bits per port register) is chip-specific. Refer to the Chip Configuration chapter to see the exact control bits for the non-identical port instance.

GPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F000	Port Data Output Register (GPIO0_PDOR)	32	R/W	0000_0000h	7.2.1/535
400F_F004	Port Set Output Register (GPIO0_PSOR)	32	W (always reads 0)	0000_0000h	7.2.2/536

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F008	Port Clear Output Register (GPIO0_PCOR)	32	W (always reads 0)	0000_0000h	7.2.3/536
400F_F00C	Port Toggle Output Register (GPIO0_PTOR)	32	W (always reads 0)	0000_0000h	7.2.4/537
400F_F010	Port Data Input Register (GPIO0_PDIR)	32	R	0000_0000h	7.2.5/537
400F_F040	Port Data Output Register (GPIO1_PDOR)	32	R/W	0000_0000h	7.2.1/535
400F_F044	Port Set Output Register (GPIO1_PSOR)	32	W (always reads 0)	0000_0000h	7.2.2/536
400F_F048	Port Clear Output Register (GPIO1_PCOR)	32	W (always reads 0)	0000_0000h	7.2.3/536
400F_F04C	Port Toggle Output Register (GPIO1_PTOR)	32	W (always reads 0)	0000_0000h	7.2.4/537
400F_F050	Port Data Input Register (GPIO1_PDIR)	32	R	0000_0000h	7.2.5/537
400F_F080	Port Data Output Register (GPIO2_PDOR)	32	R/W	0000_0000h	7.2.1/535
400F_F084	Port Set Output Register (GPIO2_PSOR)	32	W (always reads 0)	0000_0000h	7.2.2/536
400F_F088	Port Clear Output Register (GPIO2_PCOR)	32	W (always reads 0)	0000_0000h	7.2.3/536
400F_F08C	Port Toggle Output Register (GPIO2_PTOR)	32	W (always reads 0)	0000_0000h	7.2.4/537
400F_F090	Port Data Input Register (GPIO2_PDIR)	32	R	0000_0000h	7.2.5/537
400F_F0C0	Port Data Output Register (GPIO3_PDOR)	32	R/W	0000_0000h	7.2.1/535
400F_F0C4	Port Set Output Register (GPIO3_PSOR)	32	W (always reads 0)	0000_0000h	7.2.2/536
400F_F0C8	Port Clear Output Register (GPIO3_PCOR)	32	W (always reads 0)	0000_0000h	7.2.3/536
400F_F0CC	Port Toggle Output Register (GPIO3_PTOR)	32	W (always reads 0)	0000_0000h	7.2.4/537
400F_F0D0	Port Data Input Register (GPIO3_PDIR)	32	R	0000_0000h	7.2.5/537
400F_F100	Port Data Output Register (GPIO4_PDOR)	32	R/W	0000_0000h	7.2.1/535
400F_F104	Port Set Output Register (GPIO4_PSOR)	32	W (always reads 0)	0000_0000h	7.2.2/536

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F108	Port Clear Output Register (GPIO4_PCOR)	32	W (always reads 0)	0000_0000h	7.2.3/536
400F_F10C	Port Toggle Output Register (GPIO4_PTOR)	32	W (always reads 0)	0000_0000h	7.2.4/537
400F_F110	Port Data Input Register (GPIO4_PDIR)	32	R	0000_0000h	7.2.5/537

7.2.1 Port Data Output Register (GPIOx_PDOR)

This register configures the logic levels that are driven on each general-purpose output pins.

NOTE

Do not modify pin configuration registers associated with pins not available in your selected package. All un-bonded pins not available in your package will default to DISABLE state for lowest power consumption.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PDOR field descriptions

Field	Description
31–0 PDO	Port Data Output Register bits for un-bonded pins return a undefined value when read. 0 Logic level 0 is driven on pin, provided pin is configured for general-purpose output. 1 Logic level 1 is driven on pin, provided pin is configured for general-purpose output.

7.2.2 Port Set Output Register (GPIOx_PSOR)

This register configures whether to set the fields of the PDOR.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTSO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PSOR field descriptions

Field	Description
31–0 PTSO	Port Set Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to logic 1.

7.2.3 Port Clear Output Register (GPIOx_PCOR)

This register configures whether to clear the fields of PDOR.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTCO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PCOR field descriptions

Field	Description
31–0 PTCO	Port Clear Output Writing to this register will update the contents of the corresponding bit in the Port Data Output Register (PDOR) as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is cleared to logic 0.

7.2.4 Port Toggle Output Register (GPIOx_PTOR)

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTTO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PTOR field descriptions

Field	Description
31–0 PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to the inverse of its existing logic state.

7.2.5 Port Data Input Register (GPIOx_PDIR)

NOTE

Do not modify pin configuration registers associated with pins not available in your selected package. All un-bonded pins not available in your package will default to DISABLE state for lowest power consumption.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PDIR field descriptions

Field	Description
31–0 PDI	Port Data Input Reads 0 at the unimplemented pins for a particular device. Pins that are not configured for a digital function read 0. If the Port Control and Interrupt module is disabled, then the corresponding bit in PDIR does not update. 0 Pin logic level is logic 0, or is not configured for use by digital function. 1 Pin logic level is logic 1.

7.3 Functional description

7.3.1 General-purpose input

The logic state of each pin is available via the Port Data Input registers, provided the pin is configured for a digital function and the corresponding Port Control and Interrupt module is enabled.

The Port Data Input registers return the synchronized pin state after any enabled digital filter in the Port Control and Interrupt module. The input pin synchronizers are shared with the Port Control and Interrupt module, so that if the corresponding Port Control and Interrupt module is disabled, then synchronizers are also disabled. This reduces power consumption when a port is not required for general-purpose input functionality.

Chapter 8
Chip Configuration

8.1 Introduction

This chapter provides details on the individual modules of the device. It includes:

- specific module-to-module interactions not necessarily discussed in the individual module chapters
- number of instances of the module in the device and their features
- number of instances of the module in the device and the differences in the features (if any)
- register differences between the multiple instances of a module

8.2 Core modules

8.2.1 Cortex-M4 Processor Core

Cortex-M4 implements the ARMv7-ME instruction set architecture (ISA): this is the Thumb2 definition - it provides compatibility with Cortex-M3 and adds significant new capabilities with DSP and SIMD extensions. The basic multiply-accumulate instructions support operations up to 32 x 32 + 64. Cortex-M4 also includes a single-precision floating-point unit (FPU), which includes an extension register file of thirty-two 32-bit floating-point data registers. Cortex-M4 complex includes the FPU and two 32-bit system bus interfaces. The device Cortex-M4 implementations include two tightly-coupled local memories and two cache memories connected to these bus interfaces although the device implementation connects to the 64-bit system bus interconnect and supports a 32-byte cache line size.

- L1 2-way set-associative 16kB Instruction cache with 32B line size length
- L1 2-way set-associative 16kB Data cache with 32B line size length

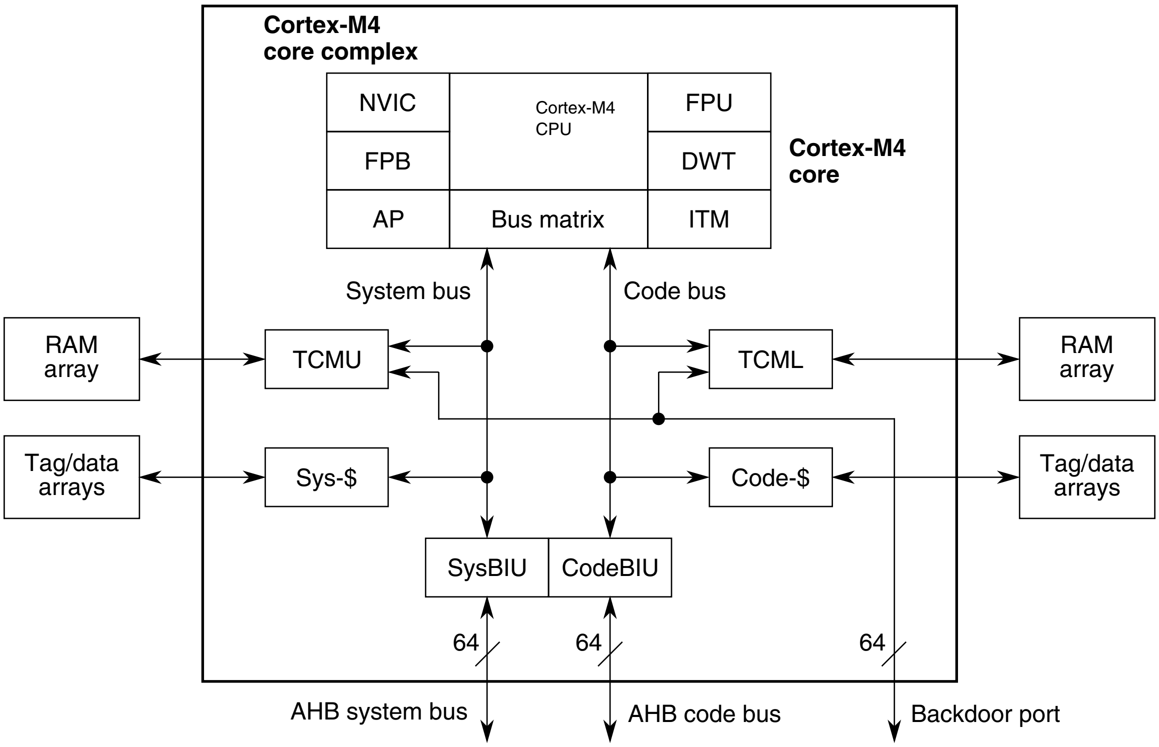


Figure 8-1. Cortex-M4 Block Diagram

Cortex-M4 core features a single issue, three stage pipeline microarchitecture. A high-level spatial pipeline block diagram of the CPU is shown below. The stages of the pipeline include:

- Fe - Instruction fetch stage where data is returned from instruction memory
- De - Instruction decode stage, generation of Load/Store Unit (LSU) address using forwarded register ports and immediate offset of LR register branch forwarding
- Ex - Instruction execute stage, single pipeline with multi-cycle stalls, LSU address/ data pipelining to AHB interface, multiply/divide and ALU with branch result

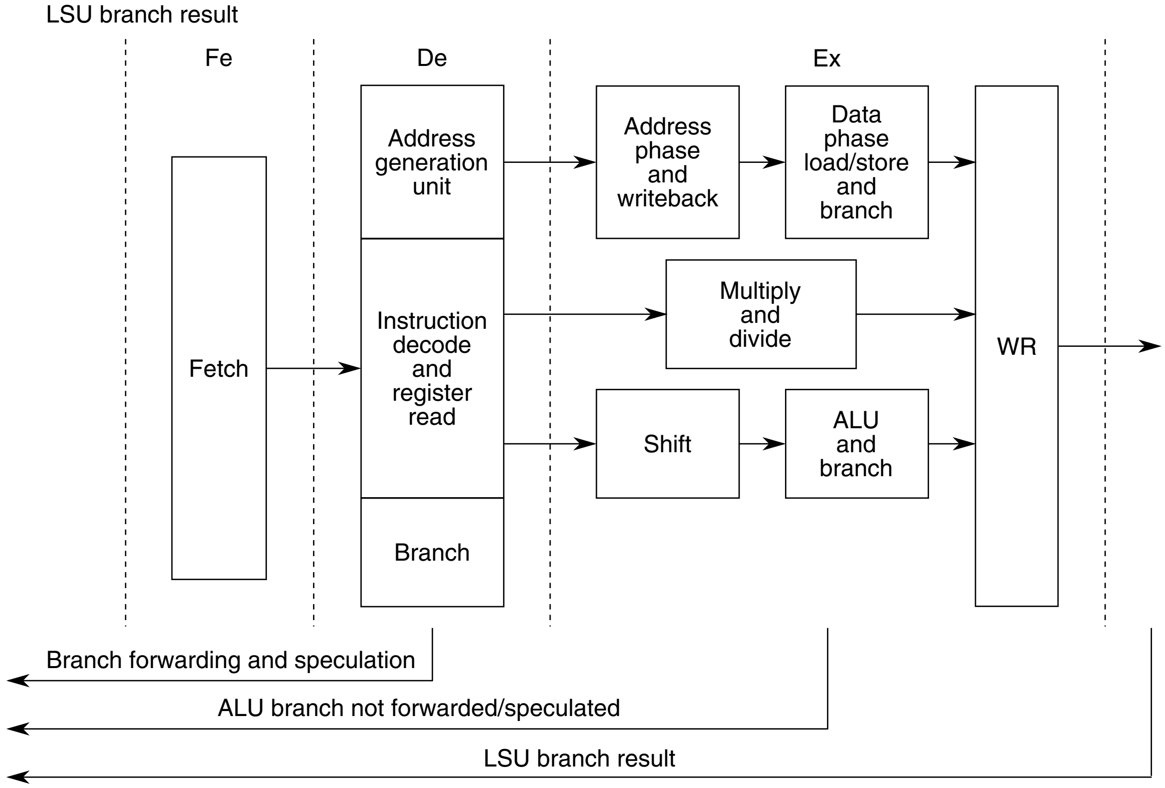


Figure 8-2. Cortex-M4 Pipeline Block Diagram

8.2.2 Cortex-M4 Instruction Fetches on the System Bus

The Cortex-M4 processors implement multiple 32-bit bus interfaces that support a Harvard memory architecture. Specifically, the cores provide a modified Harvard connection with 2-cycle pipelined AMBA-AHB code and system buses. The modified Harvard memory architecture results since the bus interfaces are activated by address range and include both instruction fetches and operand data references on a given bus port. A traditional Harvard architecture separates instruction fetches and operand data references onto specific bus ports regardless of access address.

The code bus is typically used for instruction fetching and data accesses of PC-relative data, while the system bus is typically used for operand data references to the on- and off-chip memories and peripheral accesses. This bus structure fully supports concurrent instruction fetch and data accesses, but the Cortex-M4 implementations can generate both types of references on each bus. Additionally, there is a separate 32-bit Private Peripheral Bus (PPB) connection to several important modules (for example, the Nested Vectored Interrupt Controller) accessible to only the core. By placing the various code and data sections in the appropriate locations within the memory map, overall system performance can be maximized.

To provide a “clean timing interface” on the core's system bus, instruction and vector fetch requests to this bus are registered. This increases fetch time by an additional cycle of latency because instructions fetched from the system bus take a minimum of two cycles. This also means that back-to-back instruction fetches from the system bus are not possible.

Instruction fetch requests to the code bus are not registered. It is recommended that performance critical code be located such that it fetches from the ICode bus interface as defined by addresses < 0x2000_0000 (the system bus interface includes the addresses > = 0x2000_0000 and < 0xE000_0000 and the Private Peripheral Bus is used for addresses >= 0xE000_0000).

NOTE

In the device, the memory map includes aliased address spaces that are mapped into the ICode region for code sections that reside in the system address space. As a simple example, the DDR address space is located in the system region of the memory map, but a subset of this space is aliased so that it appears in the ICode region that instructions mapped into the DDR space can be executed as maximum performance.

8.2.3 Cortex-A5 Processor Core

The Cortex-A5 processor is a high-performance, low-power ARM macrocell with an L1 cache subsystem that provides full virtual memory capabilities. The core supports the ARMv7-A instruction set architecture (supporting 32-bit ARM plus 16- and 32-bit Thumb{-2} instructions) and the microarchitecture has been optimized for area, performance/power efficiency, scalability and flexibility. It is architecturally compatible with the Cortex-A9.

It includes both an FPU and the NEON Media Processing Engine. The Cortex-A5 Floating Point Unit (FPU) is a VFPv4-D16 implementation of the ARM v7 floating-point architecture. The unit provides floating-point computation functionality that is compliant with the ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic (IEEE 754).

The FPU includes all data processing instructions and data types in the VFPv4 architecture and fully supports single-precision and double-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations. It also provides conversions between fixed-point and floating-point data formats, and floating-point constant instructions. The FPU is tightly integrated to the Cortex-A5 processor pipeline.

The NEON Media Processing Engine (MPE) extends the capabilities of the FPU with an implementation of the ARM NEON Advanced SIMD v2 instruction set for further acceleration of media and signal processing functions. It includes an additional register set supporting a rich set of SIMD operations over 8, 16, and 32-bit integer and 32-bit floating-point data types. The Cortex-A5 implements TrustZone® Technology to ensure reliable implementation of security applications ranging from digital rights management to electronic payment.

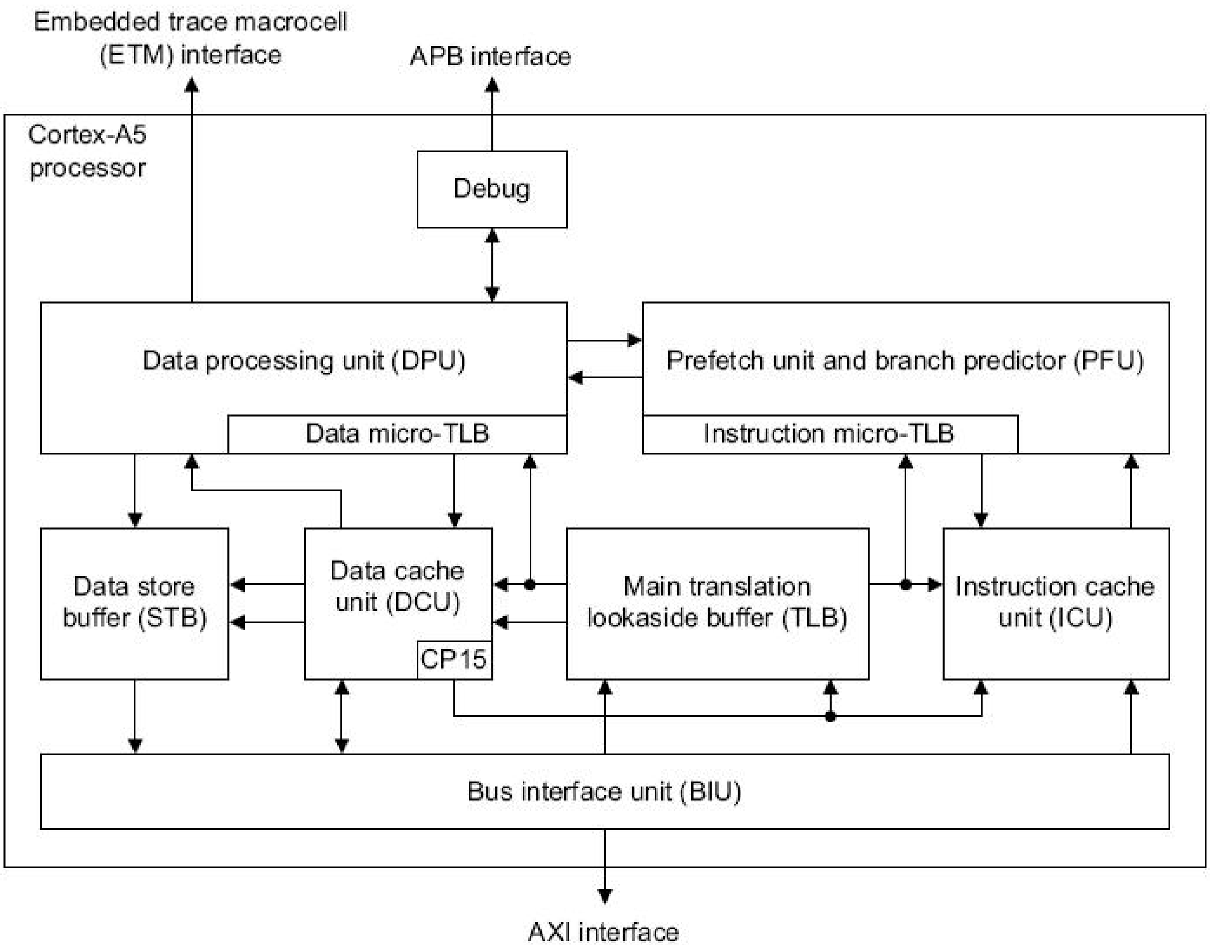


Figure 8-3. Cortex-A5 Processor Core Block Diagram

The processor microarchitecture implements a single issue, 8-stage pipeline design capable of 1.57 DMIPS per MHz performance. For the device, the configurable L1 caches are defined as 32K I- and 32K D-Caches and the system bus interface is a high performance 64-bit AXI bus that supports multiple outstanding transactions and has over 3x the memory bandwidth of the ARM1176JZ-S.

- L1 2-way set-associative 32 KB Instruction cache with 32B line size length
- L1 4-way set-associative 32 KB Data cache with 32B line size length
- L2 8-way set associative 512 KB cache with 32B line size length

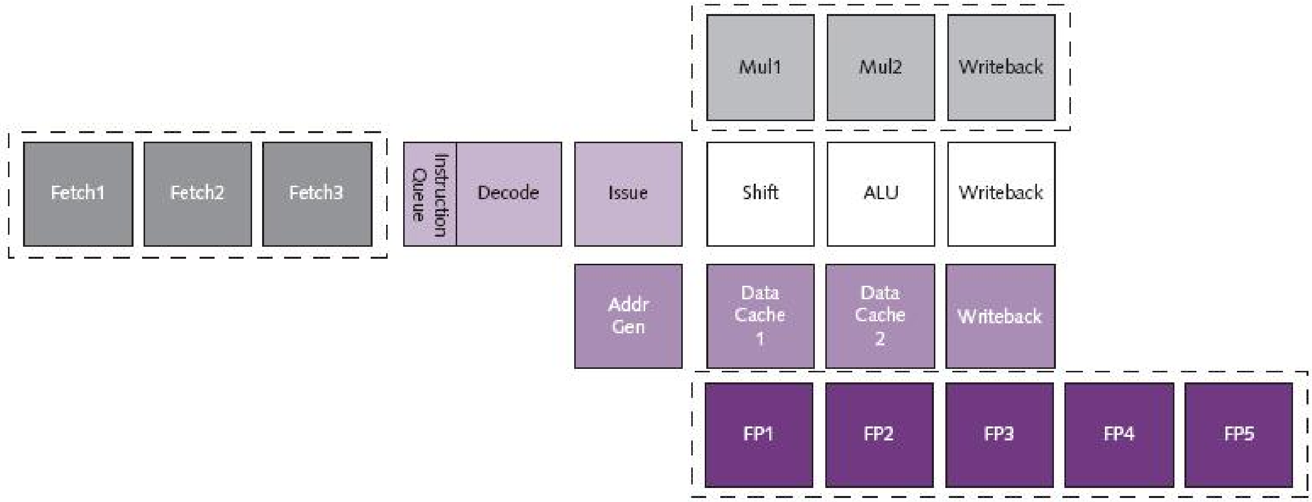


Figure 3. ARM Cortex-A5 pipelines. The basic integer pipeline is eight stages deep, including three prefetch stages that set up branch prediction. An instruction queue decouples the prefetch stages from the ALU section of the pipeline. A three-stage multiplier can execute 32- x 32-bit multiplies at a throughput rate of one instruction per cycle. Load/store instructions distribute the cache accesses over multiple stages to reduce critical timing paths. If the optional FPU or Neon extensions are present, a separate five-stage pipeline handles those operations.

Figure 8-4. Cortex-A5 Processor Pipeline Organization

NOTE

MII mode is supported for MAC0 only and when MII mode for MAC0 is enabled, RMII of MAC1 cannot be used.

8.2.4 Interrupt Assignments

Cortex-A5 uses the ARM Generic Interrupts Controller (GIC) architecture version 1.0.

Cortex-M4 uses the Nested Vectored Interrupt Controller (NVIC).

All peripheral interrupts on the device will be directed to a specific interrupt controller (GIC or NVIC) through an Interrupt Router module. In addition to the shared peripheral interrupts, there are CPU-to-CPU interrupts that will pass through the Interrupt Router. Finally, a small number of private peripheral interrupts to each core will connect directly to the NVIC or the GIC. For example, the L2 cache interrupts that are only relevant to the CortexA5 core. The GIC can support up to 16 Private Peripheral Interrupts (PPIs).

Table 8-1. Interrupt Assignment

Vector Offset Address	Cortex-M4 Vector	NVIC Interrupt ID	Name	Cortex-A5 Vector	GIC InterruptID	Type	Name
0x0000_0000	0		Initial Stack Pointer	0		h/w	Reset
0x0000_0004	1		Initial Program Counter	1		h/w	Undefined Instruction

Table continues on the next page...

Table 8-1. Interrupt Assignment (continued)

Vector Offset Address	Cortex-M4 Vector	NVIC Interrupt ID	Name	Cortex-A5 Vector	GIC InterruptID	Type	Name
0x0000_0008	2		NMI	2		h/w	Supervisor Call
0x0000_000C	3		Hard Fault	3		h/w	Prefetch Abort
0x0000_0010	4			4		h/w	Data Abort
0x0000_0014	5		Bus Fault	5		h/w	
0x0000_0018	6		Usage Fault	6	IRQ	h/w	IRQ
0x0000_001C	7			7	FIQ	h/w	FIQ
0x0000_0020	8			8		h/w	
0x0000_0024	9			9	SMC	h/w	Secure Monitor Call
0x0000_0028	10			10		h/w	
0x0000_002C	11		SVCall	11		h/w	
0x0000_0030	12		Debug Monitor	12		h/w	
0x0000_0034	13			13		h/w	
0x0000_0038	14		PendableSrvReq	14		h/w	
0x0000_003C	15		SysTick	15		h/w	
				16	0	SGI	Software-generated int
				17	1	SGI	Software-generated int
				18	2	SGI	Software-generated int
				19	3	SGI	Software-generated int
				20	4	SGI	Software-generated int
				21	5	SGI	Software-generated int
				22	6	SGI	Software-generated int
				23	7	SGI	Software-generated int
				24	8	SGI	Software-generated int
				25	9	SGI	Software-generated int
				26	10	SGI	Software-generated int
				27	11	SGI	Software-generated int

Table continues on the next page...

Table 8-1. Interrupt Assignment (continued)

Vector Offset Address	Cortex-M4 Vector	NVIC Interrupt ID	Name	Cortex-A5 Vector	GIC InterruptID	Type	Name
				28	12	SGI	Software-generated int
				29	13	SGI	Software-generated int
				30	14	SGI	Software-generated int
				31	15	SGI	Software-generated int
				32	16	PPI	Private peripheral int
				33	17	PPI	Private peripheral int
				34	18	PPI	Private peripheral int
				35	19	PPI	Private peripheral int
				36	20	PPI	Private peripheral int
				37	21	PPI	Private peripheral int
				38	22	PPI	Private peripheral int
				39	23	PPI	Private peripheral int
				40	24	PPI	Private peripheral int
				41	25	PPI	Private peripheral int
				42	26	PPI	Private peripheral int
				43	27	Global Timer	Private peripheral int
				44	28	Legacy nFIQ	Private peripheral int
				45	29	Core Timer	Private peripheral int
				46	30	Core Watchdog	Private peripheral int
				47	31	Legacy nIRQ	Private peripheral int
CPU to CPU and Directed Interrupts (CPU to CPU interrupts pass through the Interrupt Router)							
0x0000_0040	16	0	CPU to CPU int0	48	32	Peripheral Shared Interrupts (PSI)	CPU to CPU int0

Table continues on the next page...

Table 8-1. Interrupt Assignment (continued)

Vector Offset Address	Cortex-M4 Vector	NVIC Interrupt ID	Name	Cortex-A5 Vector	GIC InterruptID	Type	Name
0x0000_0044	17	1	CPU to CPU int1	49	33	PSI	CPU to CPU int1
0x0000_0048	18	2	CPU to CPU int2	50	34	PSI	CPU to CPU int2
0x0000_004C	19	3	CPU to CPU int3	51	35	PSI	CPU to CPU int3
0x0000_0050	20	4	Directed Cortex-M4(= SEMA4)	52	36	PSI	Directed Cortex-A5(= SEMA4)
0x0000_0054	21	5	Directed Cortex-M4 (= MCM)	53	37	PSI	Directed Cortex-A5 (= DBG)
0x0000_0058	22	6	Directed Cortex-M4	54	38	PSI	Directed Cortex-A5(= L2CC)
0x0000_005C	23	7	Directed Cortex-M4	55	39	PSI	Directed Cortex-A5(= PMU)
SHARED PERIPHERAL INTERRUPTS (Inputs to Interrupt Router)							
On-Platform Vectors							
0x0000_0060	24	8		56	40	DMA0	DMA transfer complete CH0-31
0x0000_0064	25	9		57	41		DMA Error Interrupt Channels 0-31
0x0000_0068	26	10		58	42	DMA1	DMA transfer complete CH0-31
0x0000_006C	27	11		59	43		DMA Error Interrupt Channels 0-31
0x0000_0070	28	12		60	44		
0x0000_0074	29	13		61	45		
0x0000_0078	30	14		62	46	MSCM-ECC0	
0x0000_007C	31	15		63	47	MSCM-ECC1	
0x0000_0080	32	16		64	48	CSU_Alarm	
0x0000_0084	33	17		65	49		
0x0000_0088	34	18		66	50	MSCM_ACTZS	All CSLn + TZASC
0x0000_008C	35	19		67	51		
Off-Platform Vectors							

Table continues on the next page...

Table 8-1. Interrupt Assignment (continued)

Vector Offset Address	Cortex-M4 Vector	NVIC Interrupt ID	Name	Cortex-A5 Vector	GIC InterruptID	Type	Name
0x0000_0090	36	20		68	52	WDOG-A5	
0x0000_0094	37	21		69	53	WDOG-M4	
0x0000_0098	38	22		70	54	WDOG-SNVS	
0x0000_009C	39	23		71	55	CP1 Boot Fail	
0x0000_00A0	40	24		72	56	QuadSPI0	
0x0000_00A4	41	25		73	57	QuadSPI1	
0x0000_00A8	42	26		74	58	DDRMC	
0x0000_00AC	43	27		75	59	SDHC0	
0x0000_00B0	44	28		76	60	SDHC1	
0x0000_00B4	45	29		77	61	Reserved	
0x0000_00B8	46	30		78	62	DCU0	
0x0000_00BC	47	31		79	63	DCU1	
0x0000_00C0	48	32		80	64	VIU	
0x0000_00C4	49	33		81	65	Reserved	
0x0000_00C8	50	34		82	66	Reserved	
0x0000_00CC	51	35		83	67	RLE	
0x0000_00D0	52	36		84	68	SEG LCD	
0x0000_00D4	53	37		85	69	Reserved	
0x0000_00D8	54	38		86	70	Reserved	
0x0000_00DC	55	39		87	71	PIT	
0x0000_00E0	56	40		88	72	LPTimer0	
0x0000_00E4	57	41		89	73	Reserved	
0x0000_00E8	58	42		90	74	FlexTimer0	
0x0000_00EC	59	43		91	75	FlexTimer1	
0x0000_00F0	60	44		92	76	FlexTimer2	
0x0000_00F4	61	45		93	77	FlexTimer3	
0x0000_00F8	62	46		94	78	Reserved	
0x0000_00FC	63	47		95	79	Reserved	
0x0000_0100	64	48		96	80	Reserved	
0x0000_0104	65	49		97	81	Reserved	
0x0000_0108	66	50		98	82	ANADIG	USBPHY 0
0x0000_010C	67	51		99	83	ANADIG	USBPHY 1
0x0000_0110	68	52		100	84	Reserved	
0x0000_0114	69	53		101	85	ADC0	
0x0000_0118	70	54		102	86	ADC1	
0x0000_011C	71	55		103	87	DAC0	
0x0000_0120	72	56		104	88	DAC1	
0x0000_0124	73	57		105	89	Reserved	

Table continues on the next page...

Table 8-1. Interrupt Assignment (continued)

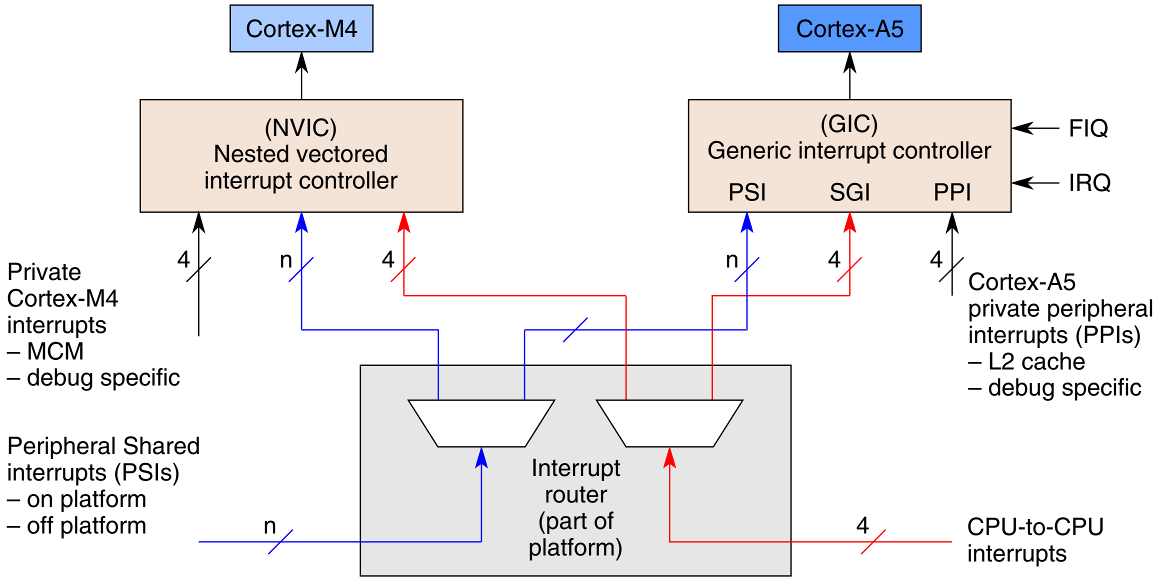
Vector Offset Address	Cortex-M4 Vector	NVIC Interrupt ID	Name	Cortex-A5 Vector	GIC InterruptID	Type	Name
0x0000_0128	74	58		106	90	FlexCAN0	
0x0000_012C	75	59		107	91	FlexCAN1	
0x0000_0130	76	60		108	92	-	
0x0000_0134	77	61		109	93	UART0	
0x0000_0138	78	62		110	94	UART1	
0x0000_013C	79	63		111	95	UART2	
0x0000_0140	80	64		112	96	UART3	
0x0000_0144	81	65		113	97	UART4	
0x0000_0148	82	66		114	98	UART5	
0x0000_014C	83	67		115	99	SPI0	
0x0000_0150	84	68		116	100	SPI1	
0x0000_0154	85	69		117	101	SPI2	
0x0000_0158	86	70		118	102	SPI3	
0x0000_015C	87	71		119	103	I2C0	
0x0000_0160	88	72		120	104	I2C1	
0x0000_0164	89	73		121	105	I2C2	
0x0000_0168	90	74		122	106	I2C3	
0x0000_016C	91	75		123	107	USBC0	
0x0000_0170	92	76		124	108	USBC1	
0x0000_0174	93	77		125	109	Reserved	
0x0000_0178	94	78		126	110	ENET0	
0x0000_017C	95	79		127	111	ENET1	
0x0000_0180	96	80		128	112	1588 Timer 0	
0x0000_0184	97	81		129	113	1588 Timer 1	
0x0000_0188	98	82		130	114	ENET Switch	
0x0000_018C	99	83		131	115	NFC	
0x0000_0190	100	84		132	116	SAI0	
0x0000_0194	101	85		133	117	SAI1	
0x0000_0198	102	86		134	118	SAI2	
0x0000_019C	103	87		135	119	SAI3	
0x0000_01A0	104	88		136	120	ESAI_BIFIFO	
0x0000_01A4	105	89		137	121	SPDIF	
0x0000_01A8	106	90		138	122	ASRC	
0x0000_01AC	107	91		139	123	VREG	HVD Interrupt
0x0000_01B0	108	92		140	124	WKPU0	
0x0000_01B4	109	93		141	125	Reserved	
0x0000_01B8	110	94		142	126	CCM	FXOSC ready interrupt

Table continues on the next page...

Table 8-1. Interrupt Assignment (continued)

Vector Offset Address	Cortex-M4 Vector	NVIC Interrupt ID	Name	Cortex-A5 Vector	GIC InterruptID	Type	Name
0x0000_01BC	111	95		143	127	CCM	Logical OR of LRF of PLL1, PLL2, PLL3 and PLL4
0x0000_01C0	112	96		144	128	SRC	
0x0000_01C4	113	97		145	129	PDB	
0x0000_01C8	114	98		146	130	EWM	
0x0000_01CC	115	99		147	131	Reserved	
0x0000_01D0	116	100		148	132	Reserved	Reserved
0x0000_01D4	117	101		149	133	Reserved	Reserved
0x0000_01D8	118	102		150	134	Reserved	
0x0000_01DC	119	103		151	135	Reserved	
0x0000_01E0	120	104		152	136	Reserved	
0x0000_01E4	121	105		153	137	Reserved	
0x0000_01E8	122	106		154	138	Reserved	
0x0000_01EC	123	107		155	139	GPIO0	Pin Interrupts / Wake-ups
0x0000_01F0	124	108		156	140	GPIO1	Pin Interrupts / Wake-ups
0x0000_01F4	125	109		157	141	GPIO2	Pin Interrupts / Wake-ups
0x0000_01F8	126	110		158	142	GPIO3	Pin Interrupts / Wake-ups
0x0000_01FC	127	111		159	143	GPIO4	Pin Interrupts / Wake-ups

The following diagram shows the high level architecture of the device interrupts.



8.3 DMA MUX Request Sources

8.3.1 DMA MUX Request Sources

This device includes a DMA request mux that allows up to 64 DMA request signals to be mapped to any of the 16 DMA channels.The first four channels of DMA MUX provide periodic triggering capability.The trigger is generated by Periodic Interrupt Timer (PIT[3:0]).

Each DMA includes two DMA request MUXes that allows up to 126 DMA request signals. As shown in Figure below, request from MUX0 can be mapped to any of the first 16 DMA channels and from MUX1, to any of the upper 16 DMA channels. While for the second DMA, it is vice versa.

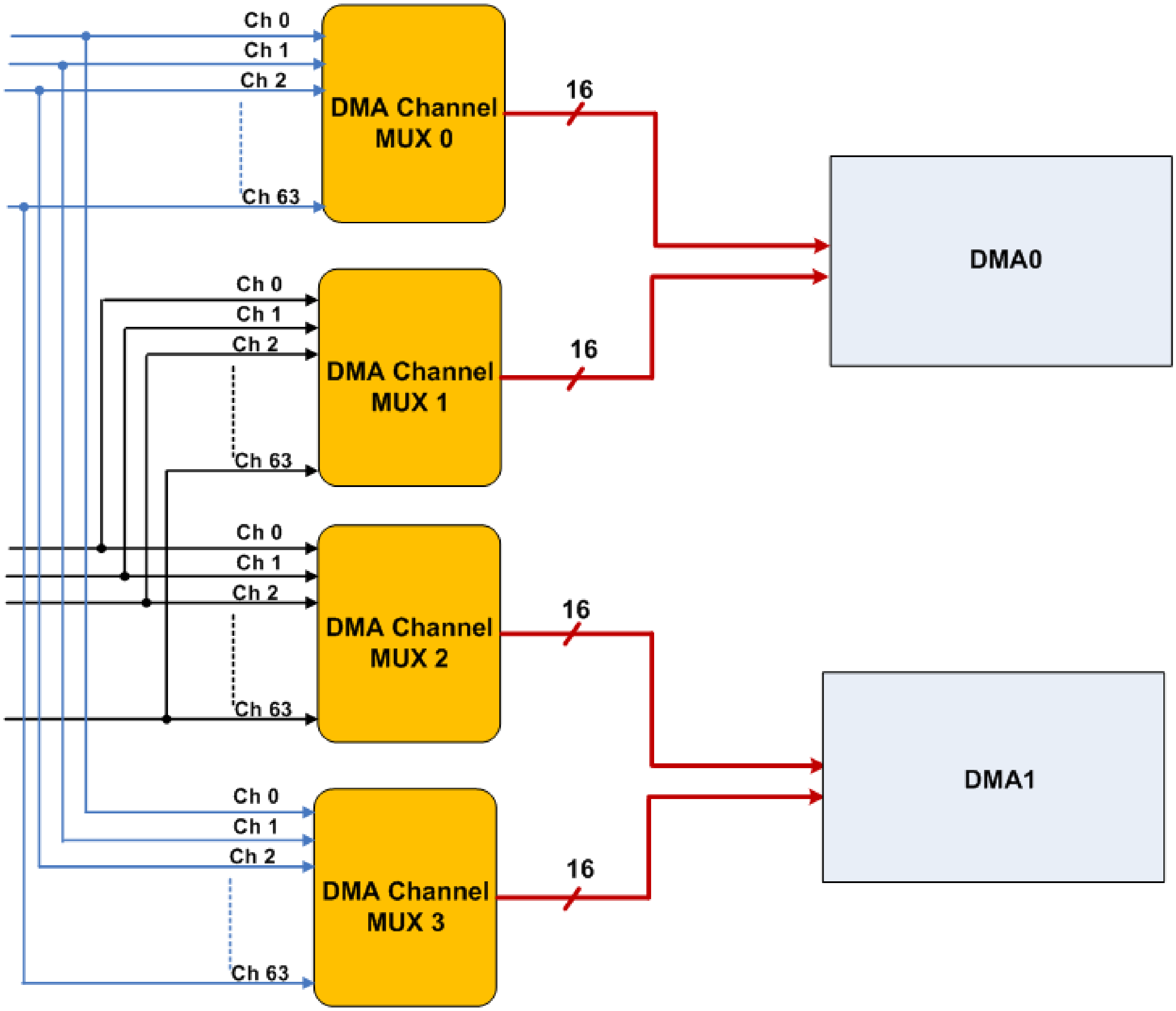


Figure 8-5. DMA Request MUX/DMA Structure

To allow for flexibility and optimal usage of the available DMA channels some of the DMA request sources are available on both muxes.

Because of the mux there is not a hard correlation between any of the DMA request sources and a specific DMA channel.

NOTE

Two duplicate Channel Mux cannot be enabled simultaneously for two (same) peripherals.

Table 8-2. DMA request sources - DMA0 (MUX 0)/DMA1 (MUX3)

Source number	Source module	Source description
0	—	Channel disabled[1]
1	Reserved	Not used
2	UART0	Receive
3	UART0	Transmit
4	UART1	Receive
5	UART1	Transmit
6	UART2	Receive
7	UART2	Transmit
8	UART3	Receive
9	UART3	Transmit
10	Reserved	—
11	Reserved	—
12	SPI0	Receive
13	SPI0	Transmit
14	SPI1	Receive
15	SPI1	Transmit
16	SAI0 (I^2S0)	Receive
17	SAI0(I^2S0)	Transmit
18	SAI1 (I^2S1)	Receive
19	SAI1 (I^2S1)	Transmit
20	SAI2 (I^2S2)	Receive
21	SAI2 (I^2S2)	Transmit
22	PDB	—
23	Reserved	—
24	FTM0	Channel 0
25	FTM0	Channel 1
26	FTM0	Channel 2
27	FTM0	Channel 3
28	FTM0	Channel 4
29	FTM0	Channel 5
30	FTM0	Channel 6
31	FTM0	Channel 7
32	FTM1	Channel 0
33	FTM1	Channel 1
34	ADC0	—
35	Reserved	—
36	QuadSPI0	—
37	Reserved	—
38	Port control module	Port A

Table continues on the next page...

Table 8-2. DMA request sources - DMA0 (MUX 0)/DMA1 (MUX3) (continued)

Source number	Source module	Source description
39	Port control module	Port B
40	Port control module	Port C
41	Port control module	Port D
42	Port control module	Port E
43	Reserved	—
44	Reserved	—
45	RLE-RX	Receive
46	RLE-TX	Transmit
47	SPDIF	Receive
48	SPDIF	Transmit
49	Reserved	
50	I2C0	Receive
51	I2C0	Transmit
52	I2C1	Receive
53	I2C1	Transmit
54	DMA MUX	Always enabled
55	DMA MUX	Always enabled
56	DMA MUX	Always enabled
57	DMA MUX	Always enabled
58	DMA MUX	Always enabled
59	DMA MUX	Always enabled
60	DMA MUX	Always enabled
61	DMA MUX	Always enabled
62	DMA MUX	Always enabled
63	DMA MUX	Always enabled

1. Configuring a DMA channel to select source 0 or any of the reserved sources disables that DMA channel.

Table 8-3. DMA request sources - DMA1 (MUX 2)/DMA0 (MUX1)

Source number	Source module	Source description
0	—	Channel disabled[1]
1	Reserved	Not used
2	UART4	Receive
3	UART4	Transmit
4	UART5	Receive
5	UART5	Transmit
6	Reserved	—
7	Reserved	—

Table continues on the next page...

Table 8-3. DMA request sources - DMA1 (MUX 2)/DMA0 (MUX1) (continued)

Source number	Source module	Source description
8	SAI3 (I^2S3)	Receive
9	SAI3 (I^2S3)	Transmit
10	SPI2	Receive
11	SPI2	Transmit
12	SPI3	Receive
13	SPI3	Transmit
14	Reserved	
15	Reserved	
16	FTM2	Channel 0
17	FTM2	Channel 1
18	FTM3	Channel 0
19	FTM3	Channel 1
20	FTM3	Channel 2
21	FTM3	Channel 3
22	FTM3	Channel 4
23	FTM3	Channel 5
24	FTM3	Channel 6
25	FTM3	Channel 7
26	ADC1	—
27	QuadSPI1	
28	Reserved	
29	Reserved	
30	Reserved	
31	Reserved	
32	DAC0	
33	DAC1	
34	ESAI_BIFIFO	Transmit
35	ESAI_BIFIFO	Receive
36	I2C2	Receive
37	I2C2	Transmit
38	I2C3	Receive
39	I2C3	Transmit
40	ASRC	1
41	ASRC	4
42	ASRC	2
43	ASRC	5
44	Reserved	
45	Reserved	
46	Reserved	

Table continues on the next page...

Table 8-3. DMA request sources - DMA1 (MUX 2)/DMA0 (MUX1) (continued)

Source number	Source module	Source description
47	Reserved	
48	Reserved	
49	Reserved	
50	Reserved	
51	Reserved	
52	ASRC	3
53	ASRC	6
54	DMA MUX	Always enabled
55	DMA MUX	Always enabled
56	DMA MUX	Always enabled
57	DMA MUX	Always enabled
58	DMA MUX	Always enabled
59	DMA MUX	Always enabled
60	DMA MUX	Always enabled
61	DMA MUX	Always enabled
62	DMA MUX	Always enabled
63	DMA MUX	Always enabled

1. Configuring a DMA channel to select source 0 or any of the reserved sources disables that DMA channel.

8.4 Wake-up Unit

8.4.1 WKPU configuration

The WKPU includes the following features.

- One NMI source
- 17 external interrupt sources with glitch filtering
- One external interrupt vector
- Five on-chip wakeup sources
- 12-bit address width
- 00000000b NMI default destination address

The table below shows the internal and external inputs to the WKUP module supported by the device.

Table 8-4. WKUP Pins

Wakeup Pin	Source	NVIC Interrupt ID	NMI
WKPU_P0	PTB0/FTM0CH0/ADC0SE2/ TRACECTL/LCD34/ SAI2_RX_BCLK/ VIU_DATA[18]/ QSPI1_A_CS0	INT92	No
WKPU_P1	PTB1/FTM0CH1/ADC0SE3/ RCON30/LCD35/ SAI2_RX_DATA/ VIU_DATA[19]/ QSPI1_A_DATA[3]	INT92	No
WKPU_P2	PTB2/FTM0CH2/ADC1SE2/ RCON31/LCD36/ SAI2_RX_SYNC/ VIU_DATA[20]/ QSPI1_A_DATA[2]	INT92	No
WKPU_P3	PTB3/FTM0CH3/ADC1SE3/ EXTRIG/LCD37/ VIU_DATA[21]/ QSPI1_A_DATA[1]	INT92	No
WKPU_P4	PTB4/FTM0CH4/SCI1_TX/ ADC0SE4/LCD38/VIU_FID/ VIU_DATA[22]/ QSPI1_A_DATA[0]	INT92	No
WKPU_P5	PTB5/FTM0CH5/SCI1_RX/ ADC1SE4/LCD39/VIU_DE/ VIU_DATA[23]/ QSPI1_A_DQS	INT92	No
WKPU_P6	PTB6/FTM0CH6/SCI1_RTS/ QSPI0_A_CS1/LCD40/ FB_CLKOUT/VIU_HSYNC/ SCI2_TX	INT92	No
WKPU_P7	PTB7/FTM0CH7/SCI1_CTS/ QSPI0_B_CS1/LCD41/ VIU_VSYNC/SCI2_RX	INT92	No
WKPU_P8	PTB11/SCI0_RX/ DCU0_TCON5/ SNVS_ALARM_OUT_B/ CKO2/ENET0_1588_TMR0	INT92	No
WKPU_P9	PTB12/SCI0_RTS/ DSPI0_PCS5/DCU0_TCON6/ FB_AD[1]/NMI/ ENET0_1588_TMR1	INT92	Yes
WKPU_P10	PTB14/CAN0_RX/I2C0_SCL/ DCU0_TCON8/DCU1_PCLK	INT92	No
WKPU_P11	PTB16/CAN1_RX/I2C1_SCL/ DCU0_TCON10	INT92	No

Table continues on the next page...

Table 8-4. WKUP Pins (continued)

Wakeup Pin	Source	NVIC Interrupt ID	NMI
WKPU_P12	PTB19/DSPI0_PCS0/ VIU_DATA[10]/CCM_OBS[1]	INT92	No
WKPU_P13	PTB20/DSPI0_SIN/LCD42/ VIU_DATA[11]/CCM_OBS[2]	INT92	No
WKPU_P14	PTB27/SAI0_RX_SYNC/ RCON22/FB_OE_b/ FB_TBST_b/NF_RE_b/ DCU1_G7	INT92	No
WKPU_P15	PTC30/SAI1_RX_SYNC/ DSPI1_PCS2/RCON28/ FB_BE0_b/FB_TSIZ0/ ADC0SE5/DCU1_B5	INT92	No
WKPU_P16	PTE20/DCU0_G7/RCON11/ LCD20/I2C0_SDA/EWM_in	INT92	No
WKPU_M0IF	LPTIMER	NA	NA
WKPU_M1IF	NA	NA	NA
WKPU_M2IF	ADC0	NA	NA
WKPU_M3IF	ADC1	NA	NA
WKPU_M4IF	PIT	NA	NA

8.5 CMU Chip Signals

8.5.1 CMU Chip Signals

This table correlate the chip-level signal name with the signal name used in the module's chapter.

Table 8-5. CMU Signals Mapping

Chip Signal Name	Module Signal Name
Bus Clock	CLKMN1f
FXOSC	CLKMN0_RMT
FIRC	CLKMT0_RMN
SIRC	CLKMT1
Tied to 0	CLKMT2

8.6 Timers

8.6.1 FlexTimer

8.6.1.1 Instantiation Information

The following table shows how these modules are configured, including the number of channels supported.

Table 8-6. FTM Instantiations

	176 LQFP	364 BGA	Features/Usage
FTM0	8	8	3-phase motor + 2 general purpose or stepper motor
FTM1	2	2	Quadrature decoder or general purpose
FTM2	2	2	Quadrature decoder or general purpose
FTM3	0	8	3-phase motor + 2 general purpose or stepper motor

The FTM1 and FTM2 configuration differs from the FTM0 and FTM3 configuration by reduced number of channels and adding Quadrature decoder.

8.6.1.2 FTM Hardware Triggers

Following table provides the hardware trigger options for FTMs.

Table 8-7. FTM Hardware Trigger options

Hardware Triggers	FTM0	FTM1	FTM2	FTM3
Trigger 1	ADC0	ADC0	ADC0	ADC0
Trigger 2	ADC1	ADC1	ADC1 or USB1 SOF	ADC1 or USB1 SOF
Trigger 3	EXTRIG or USB0 SOF	EXTRIG or USB0 SOF	EXTRIG or USB0 SOF	EXTRIG or USB0 SOF

Each flextimer has 3 trigger inputs (0/1/2) which are configured using SRC MISC0 register.

The following figures show the flextimer triggers for each instance.

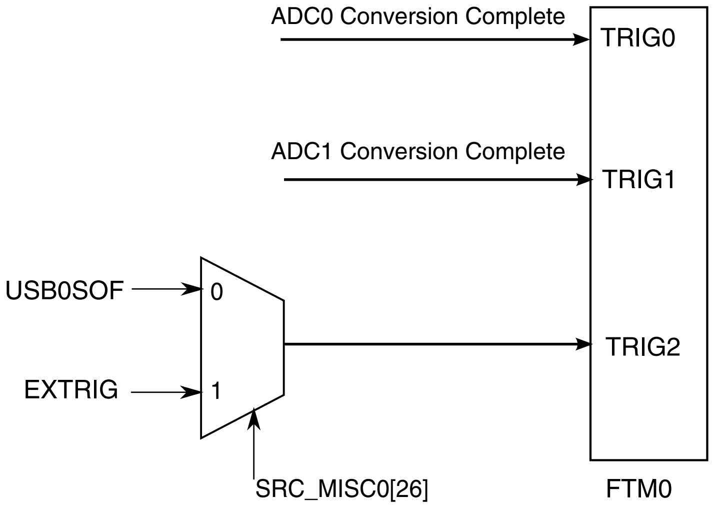


Figure 8-6. FTM0 Triggers

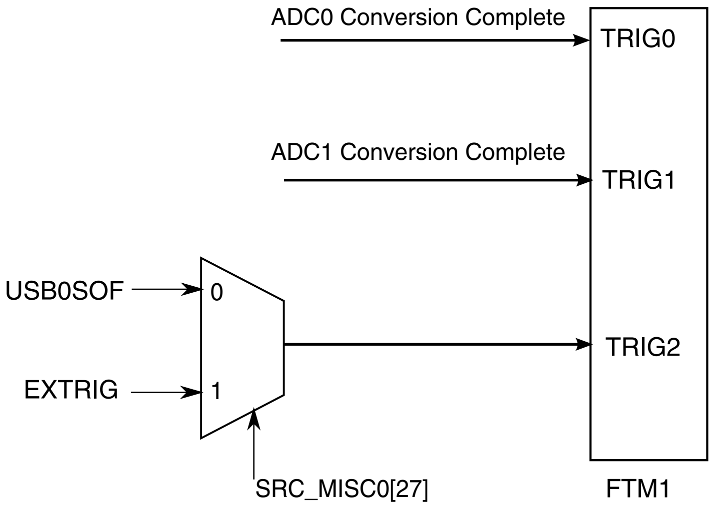


Figure 8-7. FTM1 Triggers

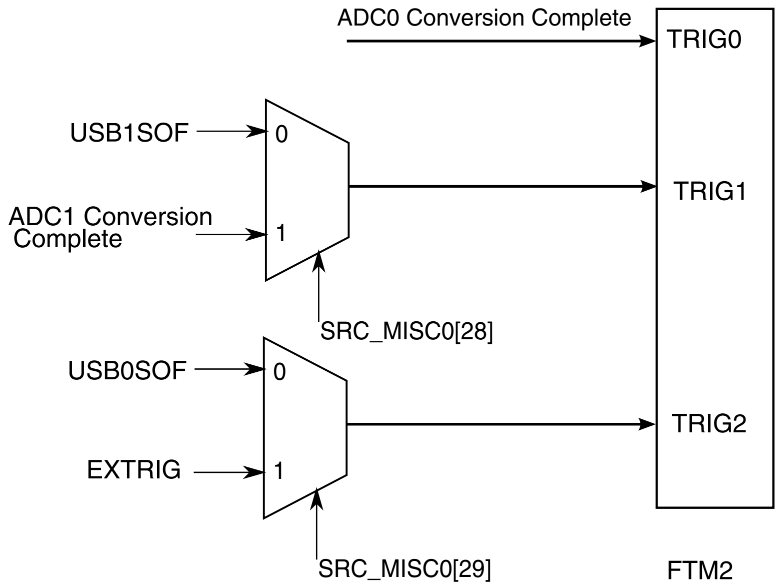


Figure 8-8. FTM2 Triggers

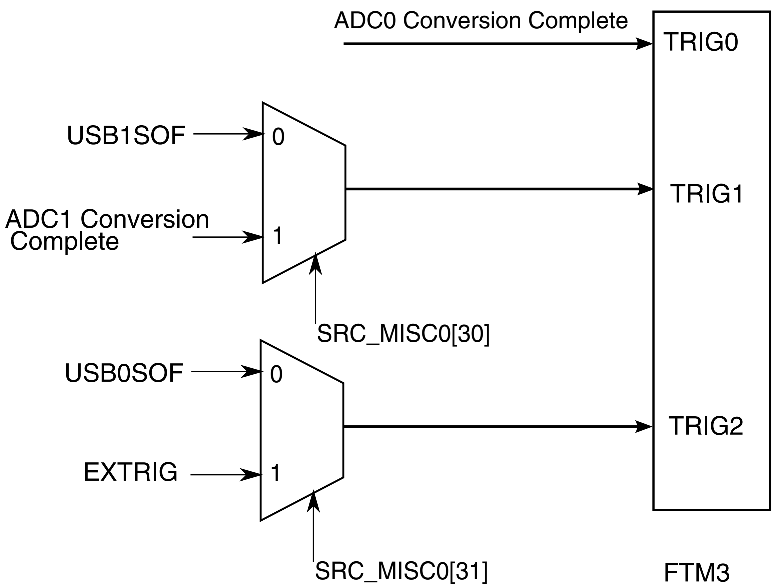


Figure 8-9. FTM3 Triggers

8.6.1.3 FTM output triggers for other modules

Following are the FTM Output Triggers options

- ADC0
- ADC1
- DAC0
- DAC1

8.6.1.4 FTM Global Time Base

This chip provides the optional FTM global time base feature (see Global time base (GTB)).

FTM0 provides the only source for the FTM global time base. The other FTM modules can share the time base as shown in the following figure:

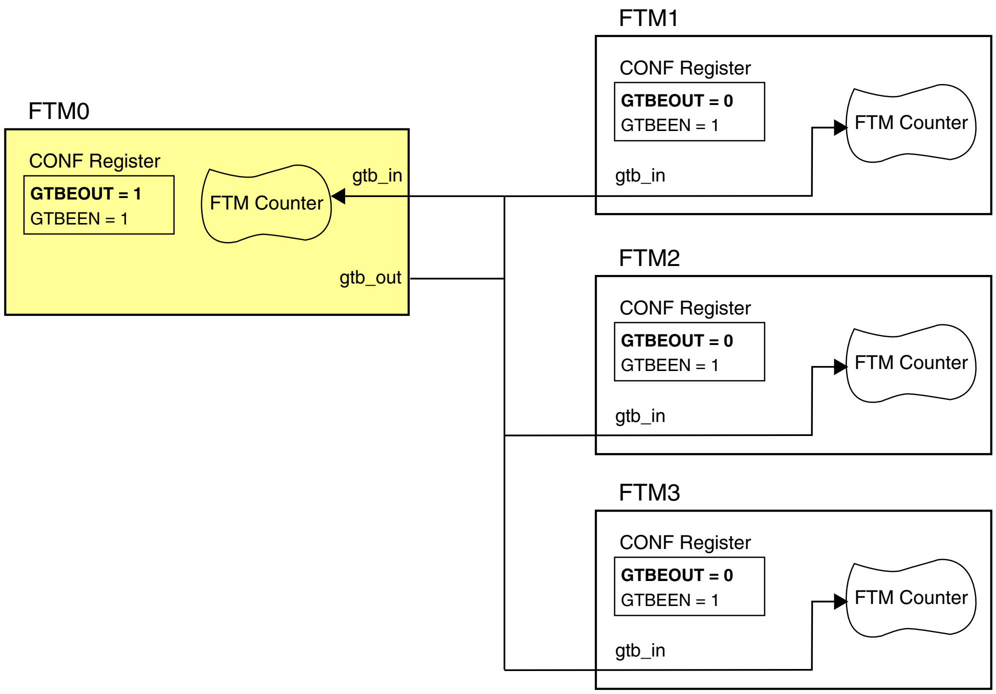


Figure 8-10. FTM Global Time Base Configuration

8.6.1.5 FTM Fault Detection Inputs

Fault inputs are useful for motor control. For motor control applications, the inputs come from a comparator. This device doesn't have an on-chip comparator; however, this device can still be used for Motor control. External interrupts (or GPIOs) from a comparator on the board can be used to indicate Fault inputs from the Motor.

In this case, the SRC would include the four Fault Inputs bits. Whenever a fault is asserted, software can write the appropriate register in SRC and this value will be driven internally to FTM.

8.6.2 Programmable Interrupt Timer(PIT)

8.6.2.1 PIT Instantiations

This device contains one PIT module with eight channels.

8.6.2.2 PIT/DMA Periodic Trigger Assignments

The PIT generates periodic trigger events to the DMA Mux as shown in the table below.

Table 8-8. PIT channel assignments for periodic DMA triggering

DMA Channel Number	PIT Channel
DMA Channel 0	PIT Channel 0
DMA Channel 1	PIT Channel 1
DMA Channel 2	PIT Channel 2
DMA Channel 3	PIT Channel 3

NOTE

When DMA channel is enabled with periodic triggering capability, and using "always enabled" DMA sources, PIT trigger is ignored.

NOTE

To implement gating the request from "always enabled" source, follow the steps below to re-assert the request:

1. Set DREQ bit in DMA_TCDn_CSR register and DMAMUX_CHCFGn[ENBL] = 0
2. Set DMAMUX_CHCFGn[ENBL] = 1, DMASREQ = channel in your DMA DONE interrupt service routine so that "always enabled" source could negate its request. Then, DMA request could be negated.

8.6.3 Programmable Delay Block (PDB)

8.6.3.1 PDB Instantiation

This device has one instance of PDB.

8.6.3.1.1 PDB Output Triggers

Table 8-9. PDB Output Triggers

PDB Parameter	Value
Number of ADC channels	2
Number of pre-triggers per ADC channel	2
Number of DAC interval triggers	2
PulseOut channels	None

8.6.3.1.2 PDB Input Trigger Connections

Table 8-10. PDB Input Trigger Options

PDB Trigger	PDB Input
0000	External Trigger at RGPIO[25]
0001	PIT Ch 0 Output
0010	PIT Ch 1 Output
0011	PIT Ch 2 Output
0100	PIT Ch 3 Output
0101	PIT Ch 4 Output
0110	PIT Ch 5 Output
0111	PIT Ch 6 Output
1000	Init and Ext Triggers from FTM0
1001	Init and Ext Triggers from FTM1
1010	Init and Ext Triggers from FTM2
1011	Init and Ext Triggers from FTM3
1101	RTC Alarm
1110	LPT Output
1111	Software Trigger

8.6.3.2 PDB Module Interconnections

PDB trigger outputs	Connection
Channel 0 triggers	ADC0 trigger
Channel 1 triggers	ADC1 trigger

8.6.3.3 DMA support on PDB

The PDB supports the DMA request functionality. One DMA request is supported.

8.6.3.4 PDB in Low-Power modes

PDB is available in the WAIT, LPRUN, UPLRUN, and STOP modes. However, the PDB is in power-gated domain and is not available in the LPSTOP1, LPSTOP2, and LPSTOP3 Low-Power Stop modes. In Low-Power Stop modes, the ADC is triggered using the Low-Power Timer (LPTMR).

8.6.3.5 PDB implementation with ADC

The following figure illustrates the implementation of the PDB interface with ADC. There is one back-to-back acknowledgement from the ADC0 to channel 0 of the PDB. In this MCU, PDB back-to-back acknowledgment connection is implemented as follows:

- PDB channel 1 pre-trigger 0 acknowledgement input: ADC0SC1B_COCO

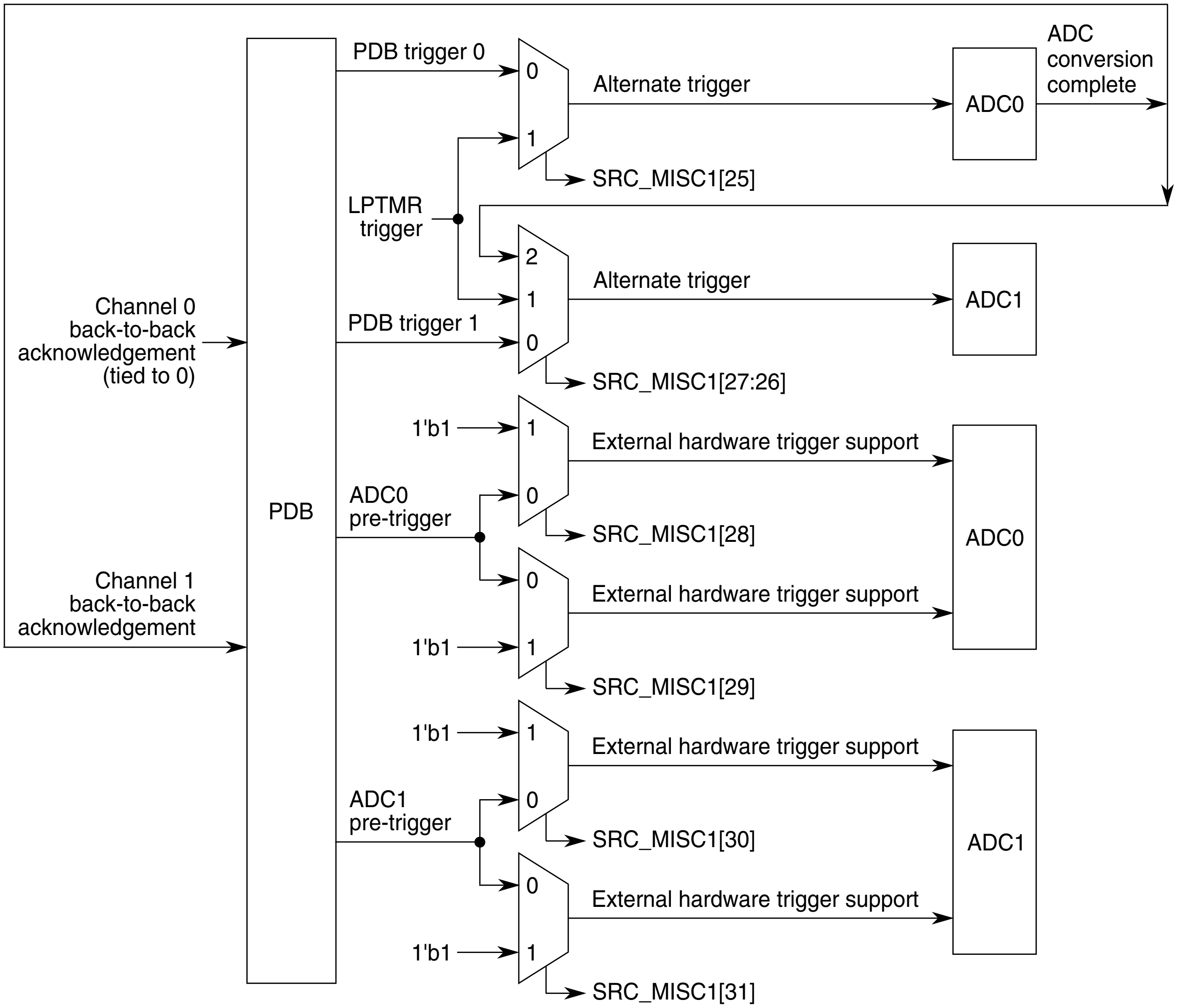


Figure 8-11. PDB implementation with ADC

8.6.4 Low-Power Timer (LPTMR)

8.6.4.1 LPTMR prescaler/glitch filter clocking options

The prescaler and glitch filter of the LPTMR module can be clocked from one of four sources determined by the LPTMR0_PSR[PCS] bitfield. The following table shows the chip-specific clock assignments for this bitfield.

NOTE

The chosen clock must remain enabled if the LPTMR is to continue operating in all required low-power modes.

LPTMR0_PSR[PCS]	Prescaler/glitch filter clock number	Chip clock
00	0	24MHz FIRC
01	1	1KHz Derived from SIRC
10	2	32KHz SXOSC
11	3	128KHz SIRC

To read about clocking on this device, refer to Clocking Overview.

8.6.5 External memory interfaces

8.6.5.1 Quad SPI

8.6.5.1.1 QuadSPI Instances

There are two instances of the QuadSPI module on this device. QuadSPI 0 supports two serial flash ports, while QuadSPI 1 supports a single serial flash port. Each QuadSPI port implements two chip-selects (CS) enabling two separate serial flashes to be attached to each individual port. The dual CS arrangement also allows dual-die packaged serial flash to be connected to a single port of a QuadSPI. The 2-port configuration of QuadSPI 0 allows two external flash devices to operate in a parallel read mode with read data from the flashes being recombined automatically in the QuadSPI module. This effectively enables an 8-bit flash interface, doubling the read bandwidth and is particularly useful for

XiP operation of fast fetch of graphics data from the external flash. Each QuadSPI4 implements a flexible 1 KB AHB buffer which can be configured in accordance with application needs.

Table 8-11. Module Instances

QuadSPI	Serial Flash Ports	# CS per Port	Parallel Mode	AHB Buffer Size	Peak DDR Read Bandwidth
QuadSPI 0	2	2	Yes	1 KB	132 MB
QuadSPI 1	1	1	No	1 KB	66 MB

QuadSPI 0 and QuadSPI 1 are available as multiplexing options on all package variants.

NOTE

Quadspi DQS Loopback mode is not supported on this device. Any reference to it in this document should be ignored.

8.6.5.1.2 QuadSPI Memory Interface

Each port of QuadSPI supports the following signals:

- SCK: Serial Clock
- IO[0:3] : Serial I/O for command, address and data
- /CS: Chip Select
- /CS2: Chip Select 2; used to select second instance of QuadSPI or select a second flash device sharing SCK and IO[0:3]
- DQS: Data strobe signal for some manufacturers for DDR read timing at 50 MHz and above

8.6.5.1.3 QuadSPI Buffer

Each QuadSPI 4 implements a flexible AHB buffer scheme to support multiple data streams the. The buffer can be partitioned into four separate buffers of different size, each associated to specific bus masters. In addition, the buffers can be associated with one or other of the connected serial flashes or both simultaneously. For details on configuration of the AHB buffers, refer to the QuadSPI chapter in this reference manual.

8.6.5.1.4 Booting from QuadSPI

Available frequency in System Boot for booting through QuadsSPI0 with multiple configuration (SDR, DDR modes):

Table 8-12. Booting from QuadSPI

	QuadSPI 0 in DDR Mode	QuadSPI 0 in SDR Mode
SCK frequency options in MHz	18	18, 60, 74

8.6.6 DRAM Controller

8.6.6.1 DDR maximum address space

The maximum address space available for the DDR controller is calculated using the following formula:

$$\text{Maximum Memory Size} = \text{Chip\_Selects} \times \text{Banks} \times 2^{\text{Address\_bits}} \times \text{Data\_Path\_Width}$$

The maximum values available for this device are:

- Chip selects = 1
- Device address = 16 rows + 10 columns = 26
- Number of banks = 8
- Memory data path width = 2 bytes

As a result, the maximum accessible memory area is 1 GB.

The address map for this configuration is shown below. Address bits 30–31 are not used. These bits are ignored when generating the address to the DRAM devices.

Bits	31 30	29	28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13	12 11 10	9 8 7 6 5 4 3 2 1	0
Field	Don't care	Chip select	Row	Bank	Column	Data path

Figure 8-12. Alternate memory map

8.6.7 Nand Flash Controller

8.6.7.1 Instantiation

The NAND Flash Controller(NFC) interfaces standard NAND Flash devices to the IC and hides the complexities of accessing the NAND Flash. It provides a glueless interface to both 8-bit and 16-bit NAND Flash parts with page sizes of 512 bytes, 2 kilobytes, 4 kilobytes and 8 kilobytes.

8.6.8 FlexBus Controller

8.6.8.1 FlexBus signal multiplexing

The multiplexing of the FlexBus address and data signals is controlled by the port control module. However, the multiplexing of some of the FlexBus control signals is controlled by the port control and FlexBus modules. The port control module registers control whether the FlexBus or another module signals are available on the external pin, while the FlexBus's CSPMCR register configures which FlexBus signals are available from the module. The control signals are grouped as illustrated:

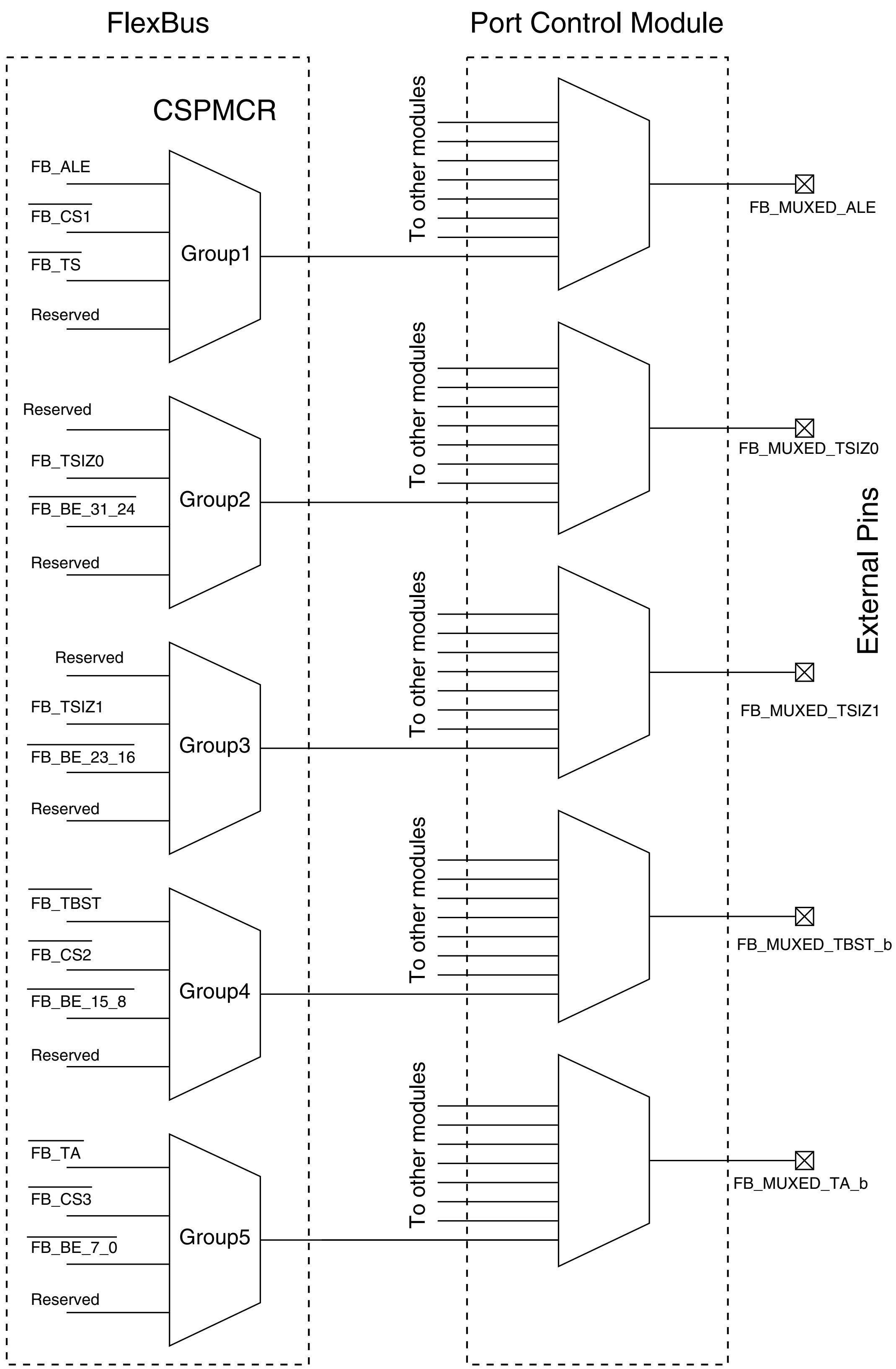


Figure 8-13. FlexBus control signal multiplexing

Therefore, use the CSPMCR and port control registers to configure which control signal is available on the external pin.

8.6.8.2 FlexBus Signal Multiplexing

The table below shows the FlexBus signal multiplexing on this device.

Table 8-13. FlexBus Multiplexing

Flexbus Signal	CSPMCR bits	Value of CSPMCR bits	Pad Connected to	Pin Connected to	ALT Mode	Signal Name
FB_ALE	[31:28]	0	PAD[93]	PTB23	ALT4	FB_MUXED_ALE
FB_CS1_b		1				
FB_TS_b		2				
Reserved		Rest				
Reserved	[27:24]	0	PAD[94]	PTB24	ALT4	FB_MUXED_TSIZ0
FB_SIZE[0]		1				
FE_BE_31_24_b		2				
Reserved		Rest				
Reserved	[23:20]	0	PAD[102]	PTC29	ALT5	FB_MUXED_TSIZ1
FB_SIZE[1]		1				
FB_BE_23_16_b		2				
Reserved		Rest				
FB_TBST_b	[19:16]	0	PAD[97]	PTB27	ALT5	FB_MUXED_TBST_b
FB_CS2_b		1				
FB_BE_15_8_b		2				
Reserved		Rest				
FB_TA_b	[15:12]	0	PAD[103]	PTC30	ALT4	FB_MUXED_BE0_b
FB_CS3_b		1				
FB_BE_7_0_b		2				
Reserved		Rest				

8.6.8.3 FlexBus External Signal

The following sections correlate the chip-level signal with the signal name used in the module's chapter.

Table 8-14. External Signals

Chip signal name	Module Signal name
FB_BE3_B	FB_BE31_24_B
FB_BE2_B	FB_BE23_16_B
FB_BE1_B	FB_BE15_8_B
FB_BE0_B	FB_BE7_0_B

8.6.8.4 FlexBus Security

FlexBus accesses are moderated by AHB TrustZone that is part of platform.

8.6.8.5 Instantiation Information

The FlexBus module is pinned out with a 32-bit multiplexed address and data bus with four chip selects. The bus allows for the configurations listed in the table below. In configurations where the full 32-bit bus is not needed those signals can be used as GPIO or other multiplexed functions.

NOTE

By default, FlexBus clocks are disabled and have to be explicitly enabled by software.

NOTE

Any access to the FlexBus device without clocks initialized may hang the bus.

Table 8-15. FlexBus Configurations

Number of Address Lines	Number of Data Lines	Number of External Signals	CSCR[bit 9] setting	Mode
32	32/16/8	32 (FB_AD[31:0])	1	Multiplexed FB_AD[31:0] allows for full 32-bit address with multiplexed 32-bit, 16-bit, or 8-bit data. 8-bit data comes out on FB_AD[7:0].

Table continues on the next page...

Table 8-15. FlexBus Configurations (continued)

Number of Address Lines	Number of Data Lines	Number of External Signals	CSCR[bit 9] setting	Mode
24 or less	16/8	24 or less (FB_AD[23:0])	1	Multiplexed FB_AD[23:0] allows for up to 24 address lines with multiplexed 16-bit or 8-bit data. 8-bit data comes out on FB_AD[7:0]
24 or less	8	32 or less (FB_AD[23:0] + FB_AD[31:24])	0	Non-multiplexed mode. FB_A[24:0] are dedicated address lines. FB_D[7:0] (FB_AD[31:24] internally) are used as dedicated data lines.
16 or less	16	32 or less (FB_AD[15:0] + FB_AD[31:16])	0	Non-multiplexed mode. FB_A[15:0] are dedicated address lines. FB_D[15:0] (FB_AD[31:16] internally) are used as dedicated data lines.
0	32/16/8	32, 16, or 8 (FB_AD[31:0], FB_AD[31:16]/ FB_AD[15:0], or FB_AD[31:24/ FB_AD[7:0])	0 or 1	Data only mode. Commonly used for interfacing to smart LCD devices. Support 32-bit, 16-bit, or 8-bt data. Either the upper or lower half of the data bus can be used as selected by CSCR[bit 9].

8.6.8.6 FlexBus Chip Select Control Register (CSCR0) Reset Value

On this device the Chip Select Control Register (CSCR0) resets to 0x003F_FC00. Configure this register as needed before performing any FlexBus acccess.

For more information about CSCR0 control bits, see Chip Select Control Register (FB_CSCR*n*).

8.6.8.7 Bus Timeout

For scenarios where auto-acknowledge feature is disabled, device can hang if there is no external FB_TA_b received. To avoid this scenario, a monitor is provided that terminates a FlexBus transaction as bus error if it does not complete within the specified time. Refer to SRC_MISC3 register description in System Reset Controller to get details about the configuration of this counter.

8.7 Communication interfaces

8.7.1 10/100 Ethernet Subsystem

8.7.1.1 Instantiation Information

Vybrid family devices will instantiate Ethernet modules as shown below.

Table 8-16. Ethernet Instantiations

	176 LQFP	364 BGA
10/100 Ethernet (MAC-NET	2	2
L2 Ethernet Switch	None	1

NOTE

1. No true AVB supported. This would be done by restricting the ethernet traffic for only Audio and Video data and managing the priority in firmware.
2. MII mode is supported for MAC0 only and when MII mode for MAC0 is enabled, RMII of MAC1 cannot be used.

8.7.1.2 MII and RMII configuration

MII is a 4-bit interface, which is clocked at 25 MHz to support 100 Mbits/sec and RMII is a 2-bit interface clock at 50 MHz to support same data rate. This device has two Ethernet MACs. MAC0 has been configured to support both RMII and MII, while MAC1 has been configured to support only RMII. Further, if MII interface on MAC0 is enabled, MAC1 cannot be used for RMII interface. Various Pins used in RMII and MII for MAC0 and the pins that can be used are described below.

Table 8-17. MII and RMII configuration for MAC0

Ethernet Signal Name	Description	Pin Configuration in MII mode	Pin Configuration in RMII mode
Transmitter Signals			
TXD0	Transmit data bit 0 (MAC to PHY)	PAD_51-ALT1	PAD_51-ALT1
TXD1	Transmit data bit 1 (MAC to PHY)	PAD_52-ALT1	PAD_52-ALT1
TXD2	Transmit data bit 2 (MAC to PHY)	PAD_76-ALT6	Not used in RMII mode
TXD3	Transmit data bit 3 (MAC to PHY)	PAD_75-ALT6	Not used in RMII mode
TXEN	Transmit data valid(MAC to PHY)	PAD_53-ALT1	PAD_53-ALT1
TXER TXCLK	Transmit data error(MAC to PHY)	PAD_77-ALT6	Not used in RMII mode
TXCLK	Transmit Clock (PHY to MAC. Can be internal for RMII as well.)	PAD_0-ALT2/ PAD2-ALT3	Both TX and RX clock in RMII mode are muxed into one signal. The source can be either: PAD_0-ALT2/ PAD2-ALT3/PLL5 main clock/Audio Ext clock as explained in Section 9.11.6 Ethernet RMII Clocking.
Receiver Signals			
RXD0	Receive bit 0 (PHY to MAC)	PAD_49-ALT1	PAD_49-ALT1
RXD1	Receive bit 1 (PHY to MAC)	PAD_48-ALT1	PAD_48-ALT1
RXD2	Receive bit 2 (PHY to MAC)	PAD_72-ALT0 (It is shared with RGPIO, so ensure IOMUX register has obe=0 for this pin when used for MII.)	Not used in RMII mode
RXD3	Receive bit 3 (PHY to MAC)	PAD_71-ALT0 (It is shared with RGPIO, so ensure IOMUX register has obe=0 for this pin when used for MII.)	Not used in RMII mode
RXEN	Receive data valid (PHY to MAC)	PAD_47-ALT1	PAD_47-ALT1
RXER	Receive data error (PHY to MAC)	PAD_50-ALT1	PAD_50-ALT1
COL	Collision (PHY to MAC)	PAD_74-ALT0 (It is shared with RGPIO, so ensure IOMUX register has obe=0 for this pin when used for MII.)	Not used in RMII mode
CRS	Carrier Sense (PHY to MAC)	PAD_73-ALT0 (It is shared with RGPIO, so ensure IOMUX register has obe=0 for this pin when used for MII.)	Not used in RMII mode.

Table continues on the next page...

Table 8-17. MII and RMII configuration for MAC0 (continued)

Ethernet Signal Name	Description	Pin Configuration in MII mode	Pin Configuration in RMII mode
RXCLK	Receive Clock (PHY to MAC. Can be internal for RMII as well.)	PAD_11-ALT0 (It is shared with RGPIO, so ensure IOMUX register has obe=0 for this pin when used for MII.)	It is muxed with TXCLK. See above for details.
MDIO	Management I/O data (Bidirectional)	PAD_46-ALT1	PAD_46-ALT1
MDC	Management Clock (PHY to MAX. Can be internal as well.)	PAD_45-ALT1	PAD_45-ALT1

8.7.1.3 IEEE 1588 Timers

The ethernet module includes a four channel timer module for IEEE 1588 timestamping. The timer supports input capture (rising, falling, or both edges), output compare (toggle or pulse with programmable polarity). The timer matches on greater than or equal (the 1588 can skip numbers, so the counter might not ever exactly match the compare value).

The counter is able to operate asynchronously to the ethernet bus by using one of four clock sources. See Clocking Chapter for more details.

8.7.1.4 Ethernet Operation in Low Power Modes

Low Power modes with Ethernet only Operation

The Ethernet module is not fully operational in any low power modes. However, the module does support magic packet detection that can generate a wakeup in low power mode if enabled.

During low power operation:

- The MAC transmit logic is disabled
- The core FIFO receive/transmit functions are disabled
- The MAC receive logic is kept in normal mode, but it ignores all traffic from the line except magic packets.

The recieve logic needed for magic packet detection is clocked using the externally-supplied RMII clock. This allows for the wakeup functionality in low power modes.

NOTE

Wakeup from Magic Packet is supported on Vybrid STOP mode but not supported on LPSTOP1/2/3 mode.

Low Power modes with Dual Ethernet and L2 Switch operation

In low power mode(STOP mode) , the ENET stops immediately and freezes operation, register values, state machines, and external pins. During this mode, the ENET clocks are shut down. Coming out of stop mode returns the ENET to operation from the state prior to stop mode entry.

NOTE

Practically this would not occur as system would not go in Stop Mode during this mode. CPU enters or executes a stop instruction for the system to go in Stop mode (low power mode) when it sees inactivity or when the system is in idle state for a long period of time. Incase there is no activity on the Ethernet Bus or no packets to forward to the other port, CPU may enter this mode. In that case, ENET should have the capability to wake up from stop mode incase of activity on Ethernet bus for the system to exit this mode (RMII clock would still be ONN to interrupt the processor on looking at any activity on the RMII bus).

Low Power modes with Dual Ethernet and L2 Switch Bypassed

In this mode, the ENET stops immediately and freezes operation, register values, state machines, and external pins. During this mode, the ENET clocks are shut down. Coming out of low power mode returns the ENET to operation from the state prior to low power mode entry.

NOTE

Any activity on the Ethernet Bus (RMII/MII interface) would wake up the system and would exist from the low power mode. Similar to the previous case, system would not enter into this mode practically incase of an ongoing activity on packet transmission via RMII interface.

Battery mode of operation

The ENET does not support any Standby Mode of operation or a capability to operate on battery incase the main supply fails. The ENET would be disabled during this Mode.

8.7.1.5 Ethernet Subsystem Interrupts

The Ethernet has multiple sources of interrupt requests. However, some of these sources are OR'd together to generate an interrupt request. See below for a summary:

Interrupt request	Interrupt source
ENET interrupt	• Transmit frame interrupt • Transmit buffer interrupt • Receive frame interrupt • Receive buffer interrupt • Wake-up • Payload receive error • Babbling receive error • Babbling transmit error • Graceful stop complete • MII interrupt – Data transfer done • Ethernet bus error • Late collision • Collision retry limit
IEEE 1588 timer interrupt	• Time stamp available • 1588 timer interrupt
Ethernet Switch interrupt	• Learning Interrupt • Output Discard for port 0 (OD0) • Output Discard for port 1 (OD1) • Output Discard for port 2 (OD2) • Receive Buffer Interrupt • Receive Frame Interrupt • Transmit Buffer Interrupt • Transmit Frame Interrupt

8.7.1.6 Ethernet switch register reset values

Some of the Ethernet Switch registers have reset values that are specific to this device. These registers and reset values are listed below and supersede the default reset values given in the Ethernet Switch chapter.

Table 8-18. Device-specific reset values

Register	Reset value
ESW_REV	0x0001_0130

8.7.2 USB 2.0 HS/FS/LS Dual Role (Host / Device) Controller

8.7.2.1 USB Configuration and Options

USB Subsystem includes the following:-

- USB 2.0 HS/FS/LS Dual Role (Host / Device) Controller (x2)
- HS 2.0 integrated PHY (x 2)

8.7.2.2 USB Host initialization and bring up

Table 8-19. Steps to start USB in Device mode

Initialization
• Power ON Vybrid. • With all the regulators activated, they generate OK. Once, all regulators are up and OK will release functional reset. • The Core works on 24 Mhz IRC. It initializes the registers, configures generic interrupt controller, and initializes interrupt handlers. • Enable 24 MHz XTAL. • Once, XTAL OK is asserted, enables all PLL (mainly SYS PLL).   • Wait for PLL Lock. (Get status from ANADIG PLL registers.) • After PLL are Locked, disable Clock gating and disable the bypass for PLL. • Switch the Clocks to PLL Clock. • Switch to USB specific configuration for running USB and initialize USB specific features with Vybrid.
Initializaing USB specific features

Table continues on the next page...

Table 8-19. Steps to start USB in Device mode (continued)

• Disable the on-chip oscillator (XOSC 24M) powerdown in next powerdown.
  • In register CCM Low Power control register (CCM_CLPCR) dessert SBYOS bit. CCM_CLPCR[SBYOS]= 0.
• Enable the USB regulator. Set bit# 0 (ENABLE_LINREG) in register ANADIG Regulator 3P0 definition register (Anadig_REG_3P0). This regulator works on 5V supply. This needs to be driven from outside the chip.
• Enable USB PLL and enable usb clocks: In ANADIG register–(Anadig_USB1_PLL_CTRL) and (Anadig_USB2_PLL_CTRL) enable EN_USB_CLKS and disable BYPASS, that is,
  • Anadig_USB1_PLL_CTRL[POWER]=1
  • Anadig_USB1_PLL_CTRL[EN_USB_CLKS]=1
  • Anadig_USB1_PLL_CTRL[BYPASS]=0
• Start the USB PHY –
  • Disable Soft reset of USB PHY – USBPHY_CTRLn[SFTRST]=0
  • Enable USB Clocks – USBPHY_CTRLn[CLKGATE]=0
  • For running the controller in LS mode, user needs to set bit#15 (ENUTMILEVEL3) and bit#14 (ENUTMILEVEL2)
  • In USB PHY Power-Down Register (USBPHY_PWD) clear all the bits and set them to zero. Anadig[USBPHY_PWD]= 0x0000_0000
• Optionally, one can enable the interrupt of USB controller going to the processor ARM Generic Interrupt Controller (GIC)-Enable Interrupt ID 107 (USB0) and Enable Interrupt ID 108 (USB1).
• The user can set the WIE bit, if the GIC bit is configured. USBC0_CTRL[WIE]=1
• Refer to “Starting USB Controller in HOST Mode”. Once, controller is set to host mode, user can start to next step.
• Setup Memory for the USB controller to start communication:
  • Configure the memory as per “Configure the USB controller to read the descriptor from the connected device and Set Address” for setting up memory for starting communication.
• User can wait for the device to detect, configure and connect for starting communication over USB. Refer to Detection for port status change, resetting the USB and speed of connected device.
• Once the setup is complete--device connected and configured. One can start communication over usb. Refer to Starting the communication over the USB from Configured Connected Device.

Starting USB Controller in Host Mode

• By default the USB on Vybrid is defaulted to device controller. For initializing them as a Host user has to do following steps:
  • In USB_OTG1_USBCMD register, set the RST bit
  • Keeping polling for de-assertion of this bit.
  • In USBMODE register configure CM bit, that is, bit 1 and bit 0
  • Keep polling and writing back 2’b11 on CM if bit are not set to 11.
  • Once these bits are set to 11, controller is configured to function like a host.
  • Refer to “Setting Up device controller” for setup device interrupt and periodic index .
  • If Port Power is required then refer to “Enable the port power in device Controller”

Enable the port power in Host controller
• In register Host Controller Structural Parameters (USBx_HCSPARAMS), read if implementation allows controlling the port power. If it reads back to ‘1’, indicates the port power can be controlled. In that case, we can configure bit number 12 of Port Status & Control (USBx_PORTSC1) register to enable port power.
Setting Up Host controller
• Setting Up interrupt threshold–In USB Command Register (USBx_USBCMD), set ITC bit to zero. Else set ITC bit as per system needs. • In USB Frame Index (USBx_FRINDEX), write 1 to the to FRINDEX. This register is used by the host controller to index the periodic frame list. The register updates every 125 microseconds (once each micro-frame). Moreover, this register cannot be written unless the Host Controller is in the 'Halted' state as indicated by the HCHalted bit inside USB Status Register (USBx_USBSTS).
Detection for port status change, resetting the USB and speed of connected device

Table continues on the next page...

Table 8-19. Steps to start USB in Device mode (continued)

- Everything is set, now user can wait for connection of any external device to the host controller. To sense the device attached, user can poll or configure an interrupt waiting for USB_OTG1_USBSTS to check the change in the port status.
  - Once bit2 of PCI bit is found asserted; dessert this bit by writing back 1 to the same location. This is W1C bit.
  - Poll for PORTSC0 register, bit #1, Current Status Change (CSC). If set to ‘1’, set port change is detected. Now clear this bit by writing back ‘1’ into the same location.
  - Check the connect status—Bit#0 of register OTG_PORTSC0. If found asserted port is connected. Else disconnect or under enumeration.
  - Check if UTMI clocks are OK in USBNC registers – USBCx_PHY – bit# 31.
  - Reset the USB port
    - IN USB Command Register (USBx_USBCMD), set RS bit#0. To start the controller.
    - Check bit#12 in USBSTS register – HCH. If set to 1, controller is in halted state.
  - Set bit#8 of PORTSC0 register for starting the reset sequence. Once this bit is written with 1, controller will start reset sequence (CHRIP) for that port.
    - Wait for USB_PORTSC[PR] bit to clear.
    - Now check for bit#2 of PORTSC register to check if the device is not disconnected due to reset sequence. If not, move to next step, else restart from “Detection for port change”.
    - Now check bit#27:26 of PORTSC register to check the port speed. If PSPD is found to be:
      - 00 – Full speed
      - 01 – Low speed
      - 10 – High speed

      If found not connected, restart from “Detection for port change”.

- Configure the host controller to create channel between software running at processor and the device connected.
  - The populated data structure for communication between the software and the device is kept in the memory. This structure needs to be kept at 64 byte aligned boundary address. The base address of this data structure needs to be loaded inside the controller to operate.
  - This address needs to be loaded at USB_OTG1_ASYNCLISTADDR.

Configure the USB controller to read the descriptor from the connected device and Set Address

- Refer the EHCI Documentation for detailed configuration and running the USB. The method shown here is one of the method for Writing Software for running the USB on Vybrid.
- USB memory need to be initialized to get configured, check features and set address for starting communication over USB.
  - Populate the Asynchronous Queue in Memory.
    - Get device feature: Reading control descriptor \ Communicate to EP0, ADD0, read Descriptor.
    - Assign Address: If readback features are supported, and need not to suspend the USB port then assign Address.
    - Once address assignment is done, populate to new QTHead where further communication to happen with new address.
    - Read feature descriptor allocated to other endpoint of the connected device.
    - Once, all the device feature are read, firmware can communicate with device accordingly to access and control different feature by populating qTD for the feature.
- The user needs to populate “Periodic framelist ” for isochronous and interrupt transfers and “asynchronous list” for control and bulk transfers.
  - For running the controller in FS mode, user needs to set endpoint speed to FS in endpoint control populated in the memory. For LS mode, user need to set the endpoint to LS in endpoint control populated in the memory.
  - User has to dump the periodic framelist baseaddress into Frame List Base Address (USBx_PERIODICLISTBASE) register .
  - For Async transfer, user has to dump async framelist base address into Asynch. Address (USBx_ASYNCLISTADDR)
  - Please refer section " Host Data Structures” for more information on Periodic and Asynchronous Framelist.

- Now, controller waits for any device to connect. Once there is any port status change, controller will generate interrupt to the processor.
- In register USBSTS, poll for USB Status, bit#0 for completion of command list given through Asynchronous queue.

Table continues on the next page...

Table 8-19. Steps to start USB in Device mode (continued)

Starting the communication over the USB from Configured Connected Device

• Refer the EHCI documentation for detailed configuration and running the USB. The method shown here is one of the method for Writing Software for running the USB on Vybrid.
• Read the “Host Data Structures” Vybrid RM.
• User needs to set following:
    • Programming the endpoint capabilities/characteristics in the queue head with the capabilities of the endpoint and device to communicate. User can program following things in the register: Endpoint number, Endpoint speed, Endpoint max packet length, and device address .
    • Program the page offset in the queue head. Each page carries a packet to be transmitted. User can actually check the status of the transmission by looking into “Current qTD pointer”, “Next qTD pointer”, and “Alternate qTD pointer”
    • In register USBSTS, poll for USB Status, bit#0 for completion of command list given through Asynchronous queue.
• After populating the entire structure into the memory, user has to set bit#7 in qTD token status. This bit is set by the software to enable the execution of transaction by the Host controller. Keep polling the remaining bits [6:0] for different communication issues.

8.7.2.3 SOF for USB Audio

This is mainly applicable when Dual Role (Host / Device) Controller is working in Device Mode. For some of the USB Audio use-cases, require some sort of audio clock recovery capability. One way to do it is to use the Start of Frame (SOF) signal which is generated at the start of a microframe in the USB 2.0 HS. This is a signal with a rate of 125 microseconds. When operating in Full Speed mode, the SOF signal has a rate of 1 ms.

Pulse that asserts for 64 system clock cycles when the SOF token is detected on the USB bus when the USB controller is in device mode.

In order to properly support USB Audio Isochronous Asynchronous mode of operation, it is necessary to measure how many audio sample clock ticks occur between two consecutive occurrences of the SOF signal. This measurement is used to provide feedback to the USB audio source in order to speed up or slow down the audio sample delivery over the USB bus.

This is the method of estimating the ratio between the USB Host clock (SOF occurrences) and the local audio clock.

Figure below shows the USB SOF connectivity with FlexTimer to enable this scheme.

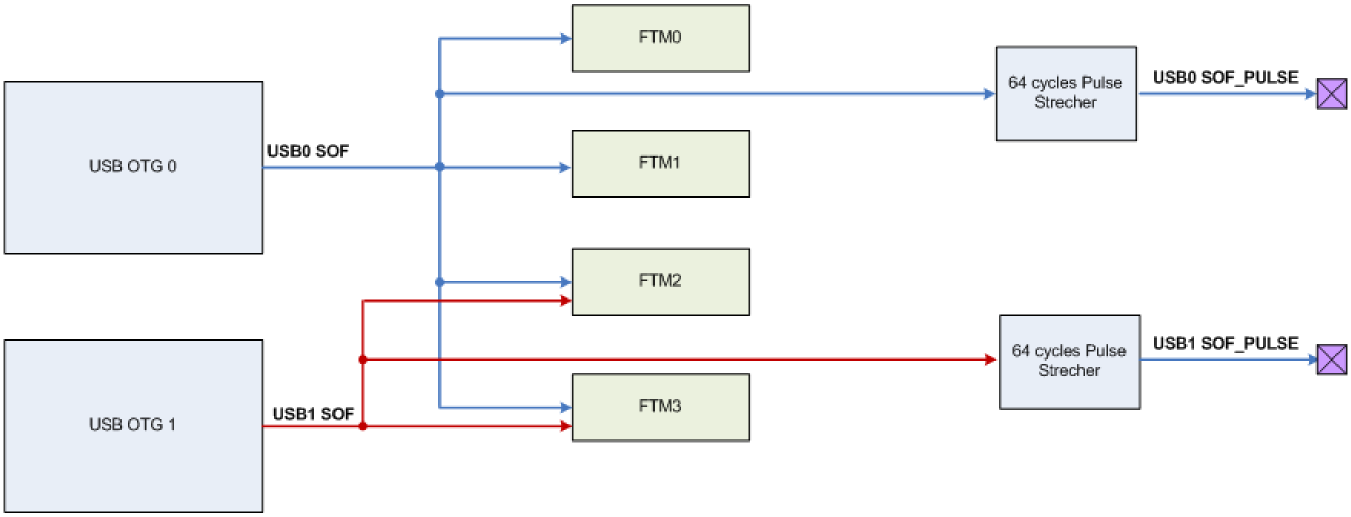


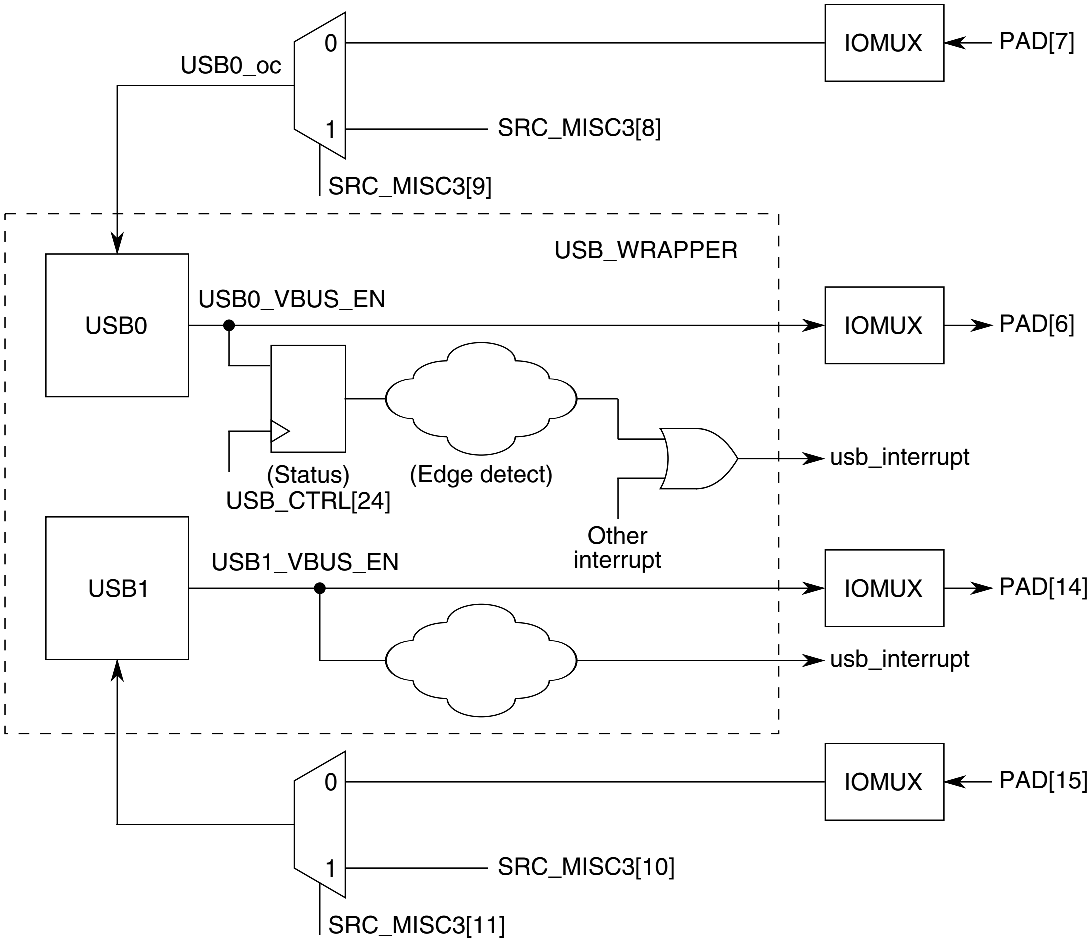
Figure 8-14. USB SOF connectivity with Flextimer

1. The two SOF signals (one from each USB port) must be brought to two timers channels of one FlexTimer. This flexibility is provided in FTM2 and FTM3 as shown in figure.
2. At least one of the SOF signals should be connected to one channel of a second FlexTimer. This will allow measuring of two sets of audio clock/SOF signals. To accomodate this USB0 SOF is connected to all Flextimers.

The audio master clock should also be provided as one of the clock options to FlexTimer.

8.7.2.4 OverCurrent and VBUS Connection

The figure below provides OverCurrent and VBUS Connection.



VBUS_EN is routed to PAD as well as an interrupt from USB. Its status is in USB_CTRL[24] and can be masked using USB_CTRL[25].

Figure 8-15. OverCurrent and VBUS Connection

8.7.3 Secure Digital Host Controller (SDHC)

8.7.3.1 SD bus pullup/pulldown constraints

The SD standard requires the SD bus signals (except the SD clock) to be pulled up during data transfers. The SDHC also provides a feature of detecting card insertion/removal, by detecting voltage level changes on DAT[3] of the SD bus. To support this DAT[3] must be pulled down. To avoid a situation where the SDHC detects voltage changes due to normal data transfers on the SD bus as card insertion/removal, the interrupt relating to this event must be disabled after the card has been inserted and detected. It can be re-enabled after the card is removed.

8.7.3.2 SDHC Wakeup

The SDHC controller is taken out of low power mode by some of its interrupts, known as wakeup interrupts. The SDHC can generate these interrupts even when clocks are not enabled. The three interrupts which can be used as wakeup events are:

1. Card Removal Interrupt
2. Card Insertion Interrupt
3. Interrupt from SDIO card

To make the interrupt a wakeup event, when all the clocks to the SDHC are disabled or when the whole system is in low power mode, the corresponding wakeup enabled bit needs to be set.

8.7.3.2.1 Setting Wake Up Events

For the SDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the CPU enters sleep mode. Before the software disables the host clock, it should ensure that all of the following conditions have been met:

- No Read or Write Transfer is active
- Data and Command lines are not active
- No interrupts are pending
- Internal data buffer is empty

8.7.3.3 SDHC Software Guidelines

SDHC interrupt does not set under the following circumstances:

1. SDHC configured to generate wakeup on card interrupt. PLL3 is selected as SDHC's baud rate clock.
2. ESDHC clocks are disabled by SDHC_SYSCTL register. For details about the register, refer to the respective chapter in this document.
3. PLLs are powered down by software before entering into STOP mode.
4. ESDHC wakeup occurs because of card interrupt.
5. Bus clock restarts but PLL does not start as software directly jumps to SDHC ISR instead of PLL enabling routine.
6. CPU reads SDHC_IRQSTAT register, but finds no bit set. The bit requires baud rate clock to get set, which needs PLL3 to get powered up.

8.7.4 UART

8.7.4.1 UART configuration information

This device contains up to six UART modules. This section describes how each module is configured on this device.

1. Standard features of all UARTs:
    - RS-485 support
    - Hardware flow control (RTS/CTS)
    - 9-bit UART to support address mark with parity
    - MSB/LSB configuration on data
2. All UARTs are clocked on the IPS bus clock. The maximum baud rate is 1/16 of related source clock frequency.
3. IrDA is available on all UARTs
4. UART0 and UART1 contains the standard features plus ISO7816
5. UART0 and UART1 contains 16-entry transmit and 16-entry receive FIFOs
6. All other UARTs contain a 8-entry transmit and 8-entry receive FIFOs
7. LON is available in UART0 and UART1
8. LIN 2.0 supported on all UARTs.

8.7.4.2 UART wakeup

The UART can be configured to generate an interrupt/wakeup on the first active edge that it receives.

8.7.4.3 UART interrupts

The UART has multiple sources of interrupt requests. However, some of these sources are OR'd together to generate a single interrupt request. See below for the mapping of the individual interrupt sources to the interrupt request:

The status interrupt combines the following interrupt sources:

Source	UART 0	UART 1	UART 2	UART 3	UART 4	UART 5
Transmit data empty	x	x	x	x	x	x

Table continues on the next page...

Source	UART 0	UART 1	UART 2	UART 3	UART 4	UART 5
Transmit complete	x	x	x	x	x	x
Idle line	x	x	x	x	x	x
Receive data full	x	x	x	x	x	x
LIN break detect	x	x	x	x	x	x
RxD pin active edge	x	x	x	x	x	x
Initial character detect	x	x	—	—	—	—

The error interrupt combines the following interrupt sources:

Source	UART 0	UART 1	UART 2	UART 3	UART 4	UART 5
Receiver overrun	x	x	x	x	x	x
Noise flag	x	x	x	x	x	x
Framing error	x	x	x	x	x	x
Parity error	x	x	x	x	x	x
Transmitter buffer overflow	x	x	x	x	x	x
Receiver buffer underflow	x	x	x	x	x	x
Transmit threshold	x	x	—	—	—	—
Receiver threshold	x	x	—	—	—	—
Wait timer	x	x	—	—	—	—
Character wait timer	x	x	—	—	—	—
Block wait timer	x	x	—	—	—	—
Guard time violation	x	x	—	—	—	—

8.7.4.4 UART Instances Register Difference

In this device, UART2, UART3, UART4, and UART5 do not support 7816 and Lon (CEA709) features. Therefore, these UART instances do not contain the respective registers. The following registers are not supported by UART2, UART3, UART4, UART5 instances in this devices.

Table 8-20. UART Instances Register Difference

Registers	UART0	UART1	UART2	UART3	UART4	UART5
UART 7816 Control Register	X	X	–	–	–	–
UART 7816 Interrupt Enable Register	X	X	–	–	–	–
UART 7816 Interrupt Status Register	X	X	–	–	–	–
UART 7816 Wait Parameter Register	X	X	–	–	–	–
UART 7816 Wait Parameter Register	X	X	–	–	–	–
UART 7816 Wait N Register	X	X	–	–	–	–
UART 7816 Wait FD Register	X	X	–	–	–	–
UART 7816 Error Threshold Register	X	X	–	–	–	–
UART 7816 Transmit Length Register	X	X	–	–	–	–
UART CEA709.1-B Packet Cycle Time Counter High	X	X	–	–	–	–
UART CEA709.1-B Packet Cycle Time Counter Low	X	X	–	–	–	–
UART CEA709.1-B Beta1 Timer (UART3_B1T)	X	X	–	–	–	–

Table continues on the next page...

Table 8-20. UART Instances Register Difference (continued)

Registers	UART0	UART1	UART2	UART3	UART4	UART5
UART CEA709.1-B Secondary Delay Timer High	X	X	–	–	–	–
UART CEA709.1-B Secondary Delay Timer Low	X	X	–	–	–	–
UART CEA709.1-B Preamble	X	X	–	–	–	–
UART CEA709.1-B Transmit Packet Length	X	X	–	–	–	–
UART CEA709.1-B Interrupt Enable Register	X	X	–	–	–	–
UART CEA709.1-B WBASE (UART3_WB)	X	X	–	–	–	–
UART CEA709.1-B Status Register	X	X	–	–	–	–
UART CEA709.1-B Status Register	X	X	–	–	–	–
UART CEA709.1-B Received Packet Length	X	X	–	–	–	–
UART CEA709.1-B Received Preamble Length	X	X	–	–	–	–
UART CEA709.1-B Collision Pulse Width	X	X	–	–	–	–

8.7.5 FlexCAN

8.7.5.1 Instantiation

There are two instances of the FlexCAN module on this device: FlexCAN0 and FlexCAN1.

Each FlexCAN is implemented with 64 message buffers each.

8.7.6 SPI

8.7.6.1 SPI Instantiation

This device contains up to 4 SPI modules.

Table 8-21. SPI Instances

SPI	176 LQFP	364 BGA
No. of SPI	3	4
SPI0 CTAR	4	4
SPI1 CTAR	2	2
SPI2 CTAR	2	2
SPI3 CTAR	N/A	2
SPI0 Chip Selects	6	6
SPI1 Chip Selects	4	4
SPI2 Chip Selects	2	2
SPI3 Chip Selects	N/A	2
TX FIFO SPI0	4	4
RX FIFO SPI0	4	4
TX FIFO SPI1	4	4
RX FIFO SPI1	4	4
TX FIFO SPI2	4	4
RX FIFO SPI2	4	4
TX FIFO SPI3	N/A	4
RX FIFO SPI3	N/A	4

NOTE

The SPI's de-serial interface (DSI) and timed serial bus (TSB) features are not supported on any SPI instance

The SPI module is clocked by the internal bus clock. The module has an internal divider, with a minimum divide of two. Thus, the SPI can run at a maximum frequency of bus clock/2.

8.7.6.2 Number of PCS

Depending on the number of DSPI instances, the number of PCS varies from 2 to 6.

For :
- DSPI0 - 6
- DSPI1 - 4
- DSPI2 - 2
- DSPI3 - 2

8.7.7 Inter-Integrated Circuit (I2C)

8.7.7.1 Instantiation Information

This device contains up to four I2C modules, I2C0, I2C1, I2C2, and I2C3.

8.8 Analog

8.8.1 12-bit Analog to Digital Converter (ADC)

8.8.1.1 ADC Instantiation

This device contains a total of 16 ADC pin inputs. The ADC signals are distributed around the chip in order to optimize the flexibility in layout and feature trade off for the user.

The ADC conversion channel is selected by setting the Pin Mux control bit.

The 176 LQFP packages do not bond out the VREFL and VREFH pins. 364 BGA offers dedicated ADC channels on pads apart from ADC channels muxed with GPIO.

Features	176 LQFP	364 BGA
No of ADC	2	2
No of Differential Channels	Not Supported	Not Supported
No of Single Ended Channels(muxed with GPIO)	6 + 6	8+8
No of dedicated single Ended Channels	None	2+2
VREFH/VREFL	None	Yes

8.8.1.2 Voltage reference selection (REFSEL settings)

The bit field definition for ADCx_CFG[REFSEL] is as follows for this chip:

00	Selects VREFH/VREFL as reference voltage. VREFH/VREFL are connected to VREFH_ADC and VREFL_ADC pads.
01	Selects VALTH/VALTL as reference voltage. VALTH/VALTL are connected to ADC reference voltage (1.2V) from Voltage Regulator.
10	Selects VBGH/VBGL as reference voltage. VBGH/VBGL are connected to ADC reference voltage (1.2V) from Voltage Regulator. (Same as 01 above).
11	Reserved

8.8.1.3 DMA Support on ADC

Applications may require continuous sampling of ADC (4K samples/sec at minimum) that may have considerable load on CPU. Though using PDB to trigger ADC may reduce some CPU load, ADC need to support DMA for higher performance where ADC is sampled at very high rate or cases were PDB is bypassed. ADC should trigger on-chip DMA (via DMA req) on conversion completion.

8.8.1.4 ADC Channel Assignments

Table 8-23. ADC Channel Assignments

ADC Channel	ADC0	ADC1	Type
0	PAD[8]	PAD[6]	External
1	PAD[9]	PAD[7]	External
2	PAD[22]	PAD[24]	External

Table continues on the next page...

Table 8-23. ADC Channel Assignments (continued)

ADC Channel	ADC0	ADC1	Type
3	PAD[23]	PAD[25]	External
4	PAD[26]	PAD[27]	External
5	PAD[103]	PAD[104]	External
6	PAD[59]	PAD[61]	External
7	PAD[60]	PAD[62]	External
8	Dedicated PAD - ADC0SE8	Dedicated PAD - ADC1SE8	External
9	Dedicated PAD - ADC0SE9	Dedicated PAD - ADC1SE9	External
10	DAC0 - external output	DAC1 - external output	Internal
11	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
12	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
13	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
14	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
15	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
16	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
17	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
18	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
19	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
20	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
21	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
22	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
23	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
24	VSS33_IO_ADC_DAC	VSS33_IO_ADC_DAC	NA
25	VREF from dedicated pad or PMU (select one based on register programming of ADC)	VREF from dedicated pad or PMU (select one based on register programming of ADC)	Internal
26	Temperature Sensor	Temperature Sensor	Internal
27	VREF from PMU	VREF from PMU	Internal
28	HI-Z	HI-Z	NA
29	HI-Z	HI-Z	NA
30	HI-Z	HI-Z	NA
31	HI-Z	HI-Z	NA

8.8.1.5 ADC interconnections

Table 8-24. ADC interconnections

Trigger Module	Trigger	Trigger
ADC	ADC0	ADC0
PDB	Trig0	Trig1
LPTIMER	LPTIMER-Trig	LPTIMER-Trig

8.8.1.6 ADC Calibration

The device fuse map (Bank7-Word5) has been assigned to store the 16-bit calibrated value for ADC0 and ADC1. This calibrated value is derived at room temperature, with nominal voltage and with max averaging. This calibrated value must be copied from Fuses to ADC register to prevent running calibration sequence at reset. Alternatively, calibration sequence supported by ADC may be run after each and every reset and the programmed value in the fuses may be ignored.

8.8.2 12-Bit Digital-to-Analog Converter (DAC)

8.8.2.1 12-bit DAC Instantiation

Table 8-25. 12-bit DAC Instantiation

	176 LQFP	364 BGA
Number of 12-bit DACs	2	2

8.8.2.2 12-bit DAC External Reference

For this device, external voltage supplied to VREFH_ADC pin is the only reference voltage for DAC reference. You need to set up DACx_STATCTRL[DACRFS]=1 to select the valid VREFH_ADC reference. When DACx_STATCTRL[DACRFS]=0, the DAC reference is connected to an internal ground node and is not a valid voltage reference. Be aware that if the DAC and ADC use the VREFH_ADC reference simultaneously,some degradation of ADC accuracy is to be expected due to DAC switching.

8.8.2.3 DMA support on DAC

Applications may require continuous sampling of DAC (4K samples/sec at minimum) that may have considerable load on CPU. The DAC supports DMA for higher performance where the DAC is sampled at a very high rate. To trigger a DMA request, the DAC uses the watermark and buffer bottom and up flags. The DAC must trigger on-chip DMA (via DMA request) on conversion completion.

8.8.2.4 DAC interconnections

Table 8-26. DAC interconnections

DAC trigger from PDB	Connection
DAC0	DAC refresh trigger 0
DAC1	DAC refresh trigger 1

8.9 Display/Video interfaces

8.9.1 Display Control Unit

8.9.1.1 Instantiation

The DCU4 modules connect to the NIC301 as bus masters.

Each DCU4 interface is a 64-bit AXI

Each DCU features an APB interface for control.

HSYNC/VSYNC, DE and Pixel Clock connect to I/O as well as to inputs of a TCON module for generation of additional timing signals

8.9.2 Timing Controller (TCON)

8.9.2.1 Instantiation

Number of TCON instances: 2 (one for each DCU)

Number of timing channels: 12

8.9.3 RLE Decoder (RLE)

8.9.3.1 Instantiation

Number of RLE_DEC instances: 1

Module is disabled by default.

8.9.4 Segmented LCD Controller

8.9.4.1 Instantiation

There is one instance of the LCD module on this device that is connected to the AIPS peripheral bridge.

8.9.4.2 LCD clock selection

The LCD module can be optionally clocked by the system bus clock, the internal 128 KHz IRC or the external 32 KHz crystal clock.

Mode	Description
Normal Run Mode	In normal run modes the LCD is clocked by the system clock.
Low Power Mode	In low power modes of operation to conserve power the LCD module is optionally clocked by the internal 128 KHz IRC or the 32 KHz external crystal oscillator. These two clock sources are available in the low power domains.

8.9.4.3 Segment LCD configuration

In this device the LCD module will be part of the Low Power Domain to enable LCD display content to be maintained. For the display to be updated the MPU must wake up from Low Power mode.

NOTE

The device does not support 5V I/O, therefore the LCD module will interface only to 3.6V glass.

Table 8-28. Number of Front and Back Planes

Parameter	Value
Number of Front Planes	40 or 38 or 36
Number of Back Planes	4 or 6 or 8

NOTE

There are only ten LCDRAM registers (LCDRAM0-LCDRAM9) implemented on this device. The other registers (LCDRAM10-LCDRAM15) that are described in the LCD chapter are not available on this device.

8.9.5 Video Interface Unit(VIU)

8.9.5.1 Instantiation

Number of VIU3 instances: 1

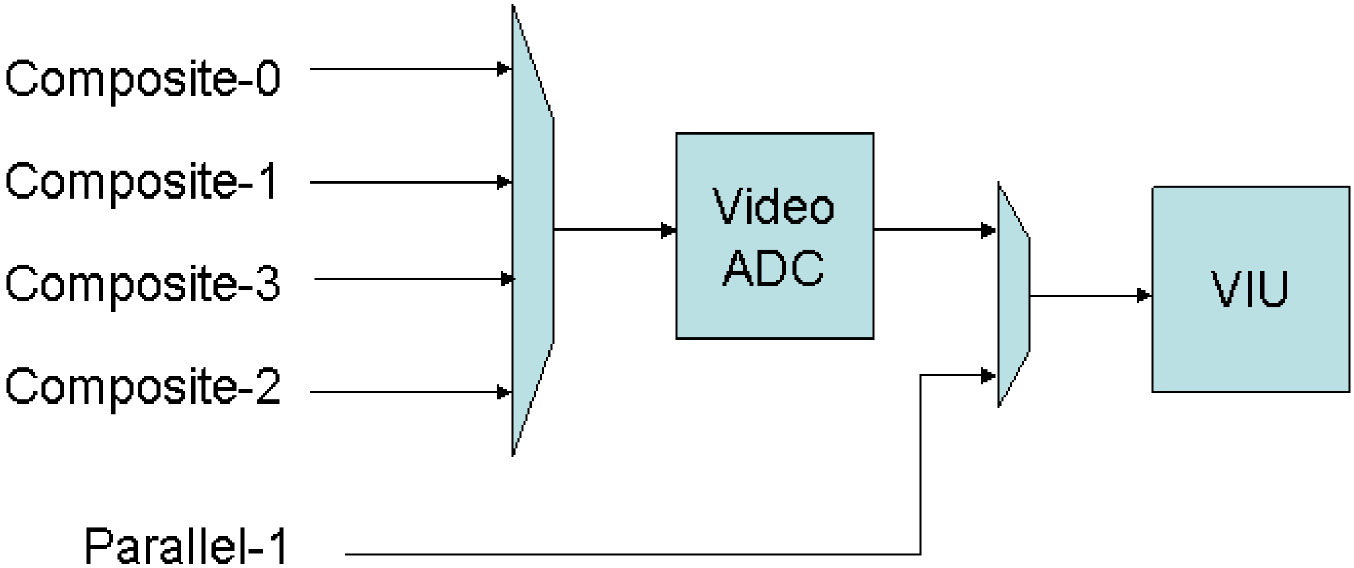
8.9.6 Video ADC(VADC)

8.9.6.1 Instantiation

This device features one instance of the VideoADC

The VideoADC accepts up to 4 single-ended analog inputs.

The VideoADC digital output interfaces directly to the VIU module as shown below.



NOTE

This mux is controlled by the MISC2 register in the SRC module.

8.10 Audio Subsystem

8.10.1 Audio Subsystem Modules

8.10.1.1 Audio Subsystem Overview

The modules that belong to the audio subsystem are the SAI0, SAI1, SAI2, SAI3, ESAI, ESAI_BIFIFO, SPDIF and ASRC. In addition, the IOMUX must be appropriately configured to get signals in and out of the chip.

SAI0-3 are synchronous audio interfaces used to transfer audio data. SAI0-3 are on the peripheral bus.

The ESAI (Enhanced Serial Audio Interface) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, SPDIF transceivers, and other processors. The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. The ESAI is connected to the IOMUX and to the ESAI_BIFIFO module.

The ESAI_BIFIFO (ESAI Bus Interface and FIFO) is the interface between the ESAI module and the shared peripheral bus. It contains the FIFOs used to buffer data to/from the ESAI, as well as providing the data word alignment and padding necessary to match the 24-bit data bus of the ESAI to the 32-bit data bus of the shared peripheral bus.

The SPDIF (Sony/Philips Digital Interface) audio module is a stereo transceiver that allows the processor to receive and transmit digital audio over it. The SPDIF receiver section includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency. A recovered clock is provided by the SPDIF receiver section and may be used to drive both internal and external components in the system. The SPDIF is connected to the shared peripheral bus.

The ASRC is a hardware audio sample rate converter supporting up to 10 channels split into 3 sampling rate pairs.

8.10.1.2 Synchronous Audio Interface (SAI)

The Synchronous Audio Interface (SAI) implements supports full-duplex serial interfaces with frame synchronization such as I2S, AC97, and codec/DSP interfaces.

This device implements four SAI modules. SAI0-2 are required for connected radio applications with Radio Tuner devices. SAI3 is added to support future expansion.

The following table shows the configuration of the SAIs:

Table 8-29. SAI Configurations

SAI Instance	Tx Data Lines	Rx Data Lines	Tx FIFO Depth	Rx FIFO Depth
SAI0	1	1	32	32
SAI1	1	1	32	32
SAI2	1	1	32	32
SAI3	1	1	64	64

8.10.1.2.1 SAI3 register details

The dedicated SAI chapter documents the identically instantiated SAI0, SAI1, and SAI2. The following table summarizes SAI3 register details that differ from the other module instances.

Table 8-30. SAI3 register differences from other SAI instances

Instance	TCR1[TFW] and RCR1[RFW]	TFR[WFP], TFR[RFP], RFR[WFP], and RFR[RFP]
SAI3	Each field is 6 bits wide	Each field is 7 bits wide

8.10.1.2.2 Simultaneous SAI DMA requests

If all four instances of the SAI module issue DMA requests simultaneously, DMA must be configured in round robin arbitration mode.

8.10.1.2.3 SAI transmitter and receiver options for MCLK selection

For an internally generated bit clock of the transmitter or receiver, all SAI instances have the same options for selecting an audio master clock (MCLK) source. TCR2[MSEL] and RCR2[MSEL] independently select the MCLK source for the transmitter and receiver, respectively. The following table shows the MSEL settings and the corresponding available MCLK sources on this device.

Table 8-31. SAIn transmitter and receiver options for MCLK selection

TCR2[MSEL] or RCR2[MSEL]	MCLK source for internally generated bit clock of transmitter or receiver
00	Bus Clock
01	Clock selected by CCM_CSCMR1[SAIn_CLK_SEL]
10	Not supported
11	Not supported

8.10.1.2.4 SAI in Stop mode

MCU wakeup from an SAI module is not supported because the SAI clock is gated in Stop mode. Before the MCU enters Stop mode, you must disable operation of the SAI transmitter and receiver in Stop mode. More specifically:

1. In the SAI, program TCSR[STOPE] and RCSR[STOPE] to 0.
2. Execute the MCU's entry to Stop mode.

8.10.1.3 Enhanced Serial Audio Interface (ESAI)

The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, SPDIF transceivers, and other processors.

The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. All serial transfers are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is also intended for periodic transfers; however, it supports up to 32 words (time

slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available.

The ESAI has 12 pins for data and clocking connection to external devices. The ESAI is internally connected to the ESAI_BIFIFO, and does not connect directly to the shared peripheral bus. The ESAI interface is designed for a 24-bit data bus, while the shared peripheral data bus is 32-bit wide. Also, the ESAI data paths are only double buffered, not allowing efficient DMA service in the applications processor environment. The ESAI_BIFIFO allows increasing the data buffering and data width matching to the shared peripheral bus.

8.10.1.4 ESAI Bus Interface and FIFO (ESAI_BIFIFO)

The ESAI_BIFIFO (ESAI Bus Interface and FIFO) is the interface between the ESAI module and the shared peripheral bus. It contains the FIFOs used to buffer data to/from the ESAI, as well as providing the data word alignment and padding necessary to match the 24-bit data bus of the ESAI to the 32-bit data bus of the shared peripheral bus..

There are two independent 128-word FIFOs, one servicing the ESAI transmit section, shared by the 6 ESAI transmitters, while the other 128-word FIFO services the ESAI receive section, shared by the 4 ESAI receivers. Each FIFO has a programmable watermark level where it can trigger a DMA service request.

The ESAI 24-bit data words are aligned and formatted according to the Transmit Word Alignment and Receive Word Alignment controls.

8.11 Miscellaneous

8.11.1 GPIO

8.11.1.1 GPIO Mapping

Table 8-32. RGPIO versus Pins

RGPIO	In GPIO module	Corresponding Pin on the chip	IOMUX register name	IOMUX register address
RGPIO[0]	PORT0[0]	PTA6	IOMUXC_PTA6	40048000
RGPIO[1]	PORT0[1]	PTA8	IOMUXC_PTA8	40048004
RGPIO[2]	PORT0[2]	PTA9	IOMUXC_PTA9	40048008

Table continues on the next page...

Table 8-32. RGPIO versus Pins (continued)

RGPIO	In GPIO module	Corresponding Pin on the chip	IOMUX register name	IOMUX register address
RGPIO[3]	PORT0[3]	PTA10	IOMUXC_PTA10	4004800C
RGPIO[4]	PORT0[4]	PTA11	IOMUXC_PTA11	40048010
RGPIO[5]	PORT0[5]	PTA12	IOMUXC_PTA12	40048014
RGPIO[6]	PORT0[6]	PTA16	IOMUXC_PTA16	40048018
RGPIO[7]	PORT0[7]	PTA17	IOMUXC_PTA17	4004801C
RGPIO[8]	PORT0[8]	PTA18	IOMUXC_PTA18	40048020
RGPIO[9]	PORT0[9]	PTA19	IOMUXC_PTA19	40048024
RGPIO[10]	PORT0[10]	PTA20	IOMUXC_PTA20	40048028
RGPIO[11]	PORT0[11]	PTA21	IOMUXC_PTA21	4004802C
RGPIO[12]	PORT0[12]	PTA22	IOMUXC_PTA22	40048030
RGPIO[13]	PORT0[13]	PTA23	IOMUXC_PTA23	40048034
RGPIO[14]	PORT0[14]	PTA24	IOMUXC_PTA24	40048038
RGPIO[15]	PORT0[15]	PTA25	IOMUXC_PTA25	4004803C
RGPIO[16]	PORT0[16]	PTA26	IOMUXC_PTA26	40048040
RGPIO[17]	PORT0[17]	PTA27	IOMUXC_PTA27	40048044
RGPIO[18]	PORT0[18]	PTA28	IOMUXC_PTA28	40048048
RGPIO[19]	PORT0[19]	PTA29	IOMUXC_PTA29	4004804C
RGPIO[20]	PORT0[20]	PTA30	IOMUXC_PTA30	40048050
RGPIO[21]	PORT0[21]	PTA31	IOMUXC_PTA31	40048054
RGPIO[22]	PORT0[22]	PTB0	IOMUXC_PTB0	40048058
RGPIO[23]	PORT0[23]	PTB1	IOMUXC_PTB1	4004805C
RGPIO[24]	PORT0[24]	PTB2	IOMUXC_PTB2	40048060
RGPIO[25]	PORT0[25]	PTB3	IOMUXC_PTB3	40048064
RGPIO[26]	PORT0[26]	PTB4	IOMUXC_PTB4	40048068
RGPIO[27]	PORT0[27]	PTB5	IOMUXC_PTB5	4004806C
RGPIO[28]	PORT0[28]	PTB6	IOMUXC_PTB6	40048070
RGPIO[29]	PORT0[29]	PTB7	IOMUXC_PTB7	40048074
RGPIO[30]	PORT0[30]	PTB8	IOMUXC_PTB8	40048078
RGPIO[31]	PORT0[31]	PTB9	IOMUXC_PTB9	4004807C
RGPIO[32]	PORT1[0]	PTB10	IOMUXC_PTB10	40048080
RGPIO[33]	PORT1[1]	PTB11	IOMUXC_PTB11	40048084
RGPIO[34]	PORT1[2]	PTB12	IOMUXC_PTB12	40048088
RGPIO[35]	PORT1[3]	PTB13	IOMUXC_PTB13	4004808C
RGPIO[36]	PORT1[4]	PTB14	IOMUXC_PTB14	40048090
RGPIO[37]	PORT1[5]	PTB15	IOMUXC_PTB15	40048094
RGPIO[38]	PORT1[6]	PTB16	IOMUXC_PTB16	40048098
RGPIO[39]	PORT1[7]	PTB17	IOMUXC_PTB17	4004809C
RGPIO[40]	PORT1[8]	PTB18	IOMUXC_PTB18	400480A0
RGPIO[41]	PORT1[9]	PTB19	IOMUXC_PTB19	400480A4

Table continues on the next page...

Table 8-32. RGPIO versus Pins (continued)

RGPIO	In GPIO module	Corresponding Pin on the chip	IOMUX register name	IOMUX register address
RGPIO[42]	PORT1[10]	PTB20	IOMUXC_PTB20	400480A8
RGPIO[43]	PORT1[11]	PTB21	IOMUXC_PTB21	400480AC
RGPIO[44]	PORT1[12]	PTB22	IOMUXC_PTB22	400480B0
RGPIO[45]	PORT1[13]	PTC0	IOMUXC_PTC0	400480B4
RGPIO[46]	PORT1[14]	PTC1	IOMUXC_PTC1	400480B8
RGPIO[47]	PORT1[15]	PTC2	IOMUXC_PTC2	400480BC
RGPIO[48]	PORT1[16]	PTC3	IOMUXC_PTC3	400480C0
RGPIO[49]	PORT1[17]	PTC4	IOMUXC_PTC4	400480C4
RGPIO[50]	PORT1[18]	PTC5	IOMUXC_PTC5	400480C8
RGPIO[51]	PORT1[19]	PTC6	IOMUXC_PTC6	400480CC
RGPIO[52]	PORT1[20]	PTC7	IOMUXC_PTC7	400480D0
RGPIO[53]	PORT1[21]	PTC8	IOMUXC_PTC8	400480D4
RGPIO[54]	PORT1[22]	PTC9	IOMUXC_PTC9	400480D8
RGPIO[55]	PORT1[23]	PTC10	IOMUXC_PTC10	400480DC
RGPIO[56]	PORT1[24]	PTC11	IOMUXC_PTC11	400480E0
RGPIO[57]	PORT1[25]	PTC12	IOMUXC_PTC12	400480E4
RGPIO[58]	PORT1[26]	PTC13	IOMUXC_PTC13	400480E8
RGPIO[59]	PORT1[27]	PTC14	IOMUXC_PTC14	400480EC
RGPIO[60]	PORT1[28]	PTC15	IOMUXC_PTC15	400480F0
RGPIO[61]	PORT1[29]	PTC16	IOMUXC_PTC16	400480F4
RGPIO[62]	PORT1[30]	PTC17	IOMUXC_PTC17	400480F8
RGPIO[63]	PORT1[31]	PTD31	IOMUXC_PTD31	400480FC
RGPIO[64]	PORT2[0]	PTD30	IOMUXC_PTD30	40048100
RGPIO[65]	PORT2[1]	PTD29	IOMUXC_PTD29	40048104
RGPIO[66]	PORT2[2]	PTD28	IOMUXC_PTD28	40048108
RGPIO[67]	PORT2[3]	PTD27	IOMUXC_PTD27	4004810C
RGPIO[68]	PORT2[4]	PTD26	IOMUXC_PTD26	40048110
RGPIO[69]	PORT2[5]	PTD25	IOMUXC_PTD25	40048114
RGPIO[70]	PORT2[6]	PTD24	IOMUXC_PTD24	40048118
RGPIO[71]	PORT2[7]	PTD23	IOMUXC_PTD23	4004811C
RGPIO[72]	PORT2[8]	PTD22	IOMUXC_PTD22	40048120
RGPIO[73]	PORT2[9]	PTD21	IOMUXC_PTD21	40048124
RGPIO[74]	PORT2[10]	PTD20	IOMUXC_PTD20	40048128
RGPIO[75]	PORT2[11]	PTD19	IOMUXC_PTD19	4004812C
RGPIO[76]	PORT2[12]	PTD18	IOMUXC_PTD18	40048130
RGPIO[77]	PORT2[13]	PTD17	IOMUXC_PTD17	40048134
RGPIO[78]	PORT2[14]	PTD16	IOMUXC_PTD16	40048138
RGPIO[79]	PORT2[15]	PTD0	IOMUXC_PTD0	4004813C
RGPIO[80]	PORT2[16]	PTD1	IOMUXC_PTD1	40048140

Table continues on the next page...

Table 8-32. RGPIO versus Pins (continued)

RGPIO	In GPIO module	Corresponding Pin on the chip	IOMUX register name	IOMUX register address
RGPIO[81]	PORT2[17]	PTD2	IOMUXC_PTD2	40048144
RGPIO[82]	PORT2[18]	PTD3	IOMUXC_PTD3	40048148
RGPIO[83]	PORT2[19]	PTD4	IOMUXC_PTD4	4004814C
RGPIO[84]	PORT2[20]	PTD5	IOMUXC_PTD5	40048150
RGPIO[85]	PORT2[21]	PTD6	IOMUXC_PTD6	40048154
RGPIO[86]	PORT2[22]	PTD7	IOMUXC_PTD7	40048158
RGPIO[87]	PORT2[23]	PTD8	IOMUXC_PTD8	4004815C
RGPIO[88]	PORT2[24]	PTD9	IOMUXC_PTD9	40048160
RGPIO[89]	PORT2[25]	PTD10	IOMUXC_PTD10	40048164
RGPIO[90]	PORT2[26]	PTD11	IOMUXC_PTD11	40048168
RGPIO[91]	PORT2[27]	PTD12	IOMUXC_PTD12	4004816C
RGPIO[92]	PORT2[28]	PTD13	IOMUXC_PTD13	40048170
RGPIO[93]	PORT2[29]	PTB23	IOMUXC_PTB23	40048174
RGPIO[94]	PORT2[30]	PTB24	IOMUXC_PTB24	40048178
RGPIO[95]	PORT2[31]	PTB25	IOMUXC_PTB25	4004817C
RGPIO[96]	PORT3[0]	PTB26	IOMUXC_PTB26	40048180
RGPIO[97]	PORT3[1]	PTB27	IOMUXC_PTB27	40048184
RGPIO[98]	PORT3[2]	PTB28	IOMUXC_PTB28	40048188
RGPIO[99]	PORT3[3]	PTC26	IOMUXC_PTC26	4004818C
RGPIO[100]	PORT3[4]	PTC27	IOMUXC_PTC27	40048190
RGPIO[101]	PORT3[5]	PTC28	IOMUXC_PTC28	40048194
RGPIO[102]	PORT3[6]	PTC29	IOMUXC_PTC29	40048198
RGPIO[103]	PORT3[7]	PTC30	IOMUXC_PTC30	4004819C
RGPIO[104]	PORT3[8]	PTC31	IOMUXC_PTC31	400481A0
RGPIO[105]	PORT3[9]	PTE0	IOMUXC_PTE0	400481A4
RGPIO[106]	PORT3[10]	PTE1	IOMUXC_PTE1	400481A8
RGPIO[107]	PORT3[11]	PTE2	IOMUXC_PTE2	400481AC
RGPIO[108]	PORT3[12]	PTE3	IOMUXC_PTE3	400481B0
RGPIO[109]	PORT3[13]	PTE4	IOMUXC_PTE4	400481B4
RGPIO[110]	PORT3[14]	PTE5	IOMUXC_PTE5	400481B8
RGPIO[111]	PORT3[15]	PTE6	IOMUXC_PTE6	400481BC
RGPIO[112]	PORT3[16]	PTE7	IOMUXC_PTE7	400481C0
RGPIO[113]	PORT3[17]	PTE8	IOMUXC_PTE8	400481C4
RGPIO[114]	PORT3[18]	PTE9	IOMUXC_PTE9	400481C8
RGPIO[115]	PORT3[19]	PTE10	IOMUXC_PTE10	400481CC
RGPIO[116]	PORT3[20]	PTE11	IOMUXC_PTE11	400481D0
RGPIO[117]	PORT3[21]	PTE12	IOMUXC_PTE12	400481D4
RGPIO[118]	PORT3[22]	PTE13	IOMUXC_PTE13	400481D8
RGPIO[119]	PORT3[23]	PTE14	IOMUXC_PTE14	400481DC

Table continues on the next page...

Table 8-32. RGPIO versus Pins (continued)

RGPIO	In GPIO module	Corresponding Pin on the chip	IOMUX register name	IOMUX register address
RGPIO[120]	PORT3[24]	PTE15	IOMUXC_PTE15	400481E0
RGPIO[121]	PORT3[25]	PTE16	IOMUXC_PTE16	400481E4
RGPIO[122]	PORT3[26]	PTE17	IOMUXC_PTE17	400481E8
RGPIO[123]	PORT3[27]	PTE18	IOMUXC_PTE18	400481EC
RGPIO[124]	PORT3[28]	PTE19	IOMUXC_PTE19	400481F0
RGPIO[125]	PORT3[29]	PTE20	IOMUXC_PTE20	400481F4
RGPIO[126]	PORT3[30]	PTE21	IOMUXC_PTE21	400481F8
RGPIO[127]	PORT3[31]	PTE22	IOMUXC_PTE22	400481FC
RGPIO[128]	PORT4[0]	PTE23	IOMUXC_PTE23	40048200
RGPIO[129]	PORT4[1]	PTE24	IOMUXC_PTE24	40048204
RGPIO[130]	PORT4[2]	PTE25	IOMUXC_PTE25	40048208
RGPIO[131]	PORT4[3]	PTE26	IOMUXC_PTE26	4004820C
RGPIO[132]	PORT4[4]	PTE27	IOMUXC_PTE27	40048210
RGPIO[133]	PORT4[5]	PTE28	IOMUXC_PTE28	40048214
RGPIO[134]	PORT4[6]	PTA7	IOMUXC_PTA7	40048218

8.11.1.2 Configuring a pin as GPIO

To program a pin as GPIO:

1. Program the corresponding register in IOMUX. For example, to configure PORT0-pin16 (PTA16), program IOMUXC_PTA16.
2. Program proper ALT mode in this register and program IBE (for Input) and OBE (for Output).
3. Program the GPIO registers (PDOR, PSOR, PCOR, PTOR, PDIR) you can set/reset this pin or read from this pin.

NOTE

For details about IOMUX and GPIO registers, see the respective chapters of this document.

8.11.1.3 Port 4 Register Differences

In this device, 135 PADs are implemented across 5 RGPIO port instances as 4 x 32 + 7. The last instance has only 7 PAD control registers. The dedicated RGPIO chapter documents the identically instantiated Port 0, 1, 2, and 3. The following table summarizes Port E register details that differ from the other module instances.

Table 8-33. Port 4 register differences from other Port instances

Instance	Port Data Output Register (GPIO_PDOR)	Port Data Input Register (GPIO_PDIR)
Port 4	PDO[6:0] field is 7 bits wide	PDI [6:0] field is 7 bits wide

Chapter 9
Clocking Overview

9.1 Introduction

The clocking on this device is flexible and allows clock from multiple sources to be used across different IPs and peripherals. Clocks on the device are controlled by ANADIG, SCSC and CCM modules. All clock sources lead to the CCM module, which generates the root clocks for the bus and peripherals of the device. The control logic in CCM additionally generates special clocks for peripherals which run asynchronously to the root clock and are termed as Auxiliary clocks. This chapter provides a top level description of the clock sources and their arrangement.

Read the following chapters in conjunction with this chapter for the respective details.

- CCM - To program the selection of clock sources, refer to the registers in Clock Controller Module (CCM) chapter. CCM also controls clock gating on this device.
- ANADIG - To program the PLLs and PFDs configuration, refer to the registers in ANADIG chapter.
- SCSC - To configure SIRC and SOSC controls, refer to the registers in SCSC chapter.

9.2 High Level Clocking Diagram

The figure below provides an overview of system clock generation on this device.

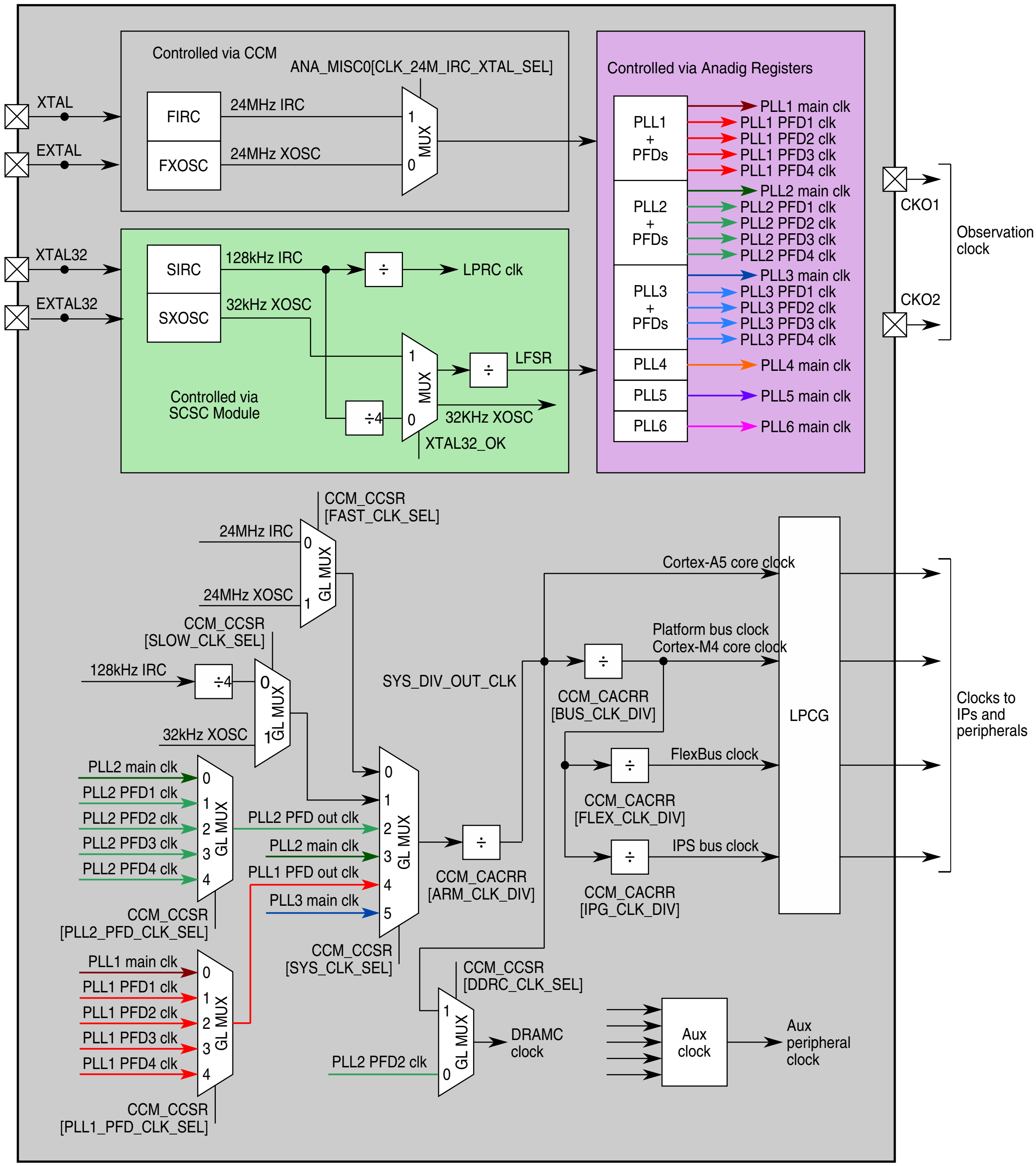


Figure 9-1. Clock Generation and Distribution

NOTE

XTAL32_OK is an internal signal signifying availability of the SXOSC clock. If the external 32 KHz XOSC signal is not available, SIRC/4 is used instead.

NOTE

LPRC CLK - 1KHz clock that can be used by LPTMR counter.

The table below defines the clocks on the device.

Table 9-1. Clock Definitions

Clock	Definition
Platform Bus/Cortex-M4 Clock[1]	Cortex-M4, Network interconnect (NIC301), and the host bus interfaces of different IPs operate on this clock.
Cortex-A5 Core clock	Cortex-A5 core clock
IPS Clock	This is the register read/write interface clock for any IP and it is a gated version of the IPG clock automatically enabled when accessing the register interface. This clock is active only during register access.
IPG Bus Clock	IPG clock is half the frequency of platform bus clock. Internal engines for IPs work on this clock. Program CCM_CACRR[IPG_CLK_DIV] to 01.
Auxillary Clock	Asynchronous clocks required by individual IPs.
LFSR Clock	Divided SXOSC clock for tamper detection. Refer to Security Architecture for details.
LPRC Clock	1 KHz clock that can be used by LPTMR counter. Refer to LPTMR_PSR register in the LPTMR chapter for details.

1. Platform Bus/Cortex-M4 Clock, Cortex-A5 Core Clock, IPS Bus clock, and IPG Bus Clock are synchronus to each other.

9.3 Clock Sources

The device implements seven PLLs and four oscillators to meet the clocking requirements of the system. Some PLL outputs are fixed, while others are configurable. Three of the PLLs have further supplemented with PFD (Phase Frequency Dividers) support, which generate independent clock frequencies using the VCO frequency generated by the PLLs. These PFD outputs can then be selected to clock the system and various modules.

In addition to the PLLs, the device also includes:

- 2 external crystal oscillators (XOSC)
    - Fast external crystal oscillator 24 MHz (FXOSC)
    - Slow external crystal oscillator 32 KHz (SXOSC)
- 2 internal RC (IRC) oscillators
    - Fast internal RC oscillator 24 MHz (FIRC)
    - Slow internal RC oscillator 128 KHz (SIRC)

NOTE

FXOSC/FIRC clocks are configured by the CCM module. SXOSC/SIRC clocks are configured by the SCSC module.

NOTE

If the external 32KHz XOSC signal is not available, the chip will automatically use the SIRC/4 clock to provide a 32KHz clock, provided the SIRC is not disabled by software.

9.4 PLL Summary

9.4.1 PLL Block DIAGRAM

The figure below provides an overview of PLL and PFD controls on this device. Only three of the PLLs have a corresponding PFD.

NOTE

PLL and PFD controls are programmed via ANADIG registers. The table below maps the bits/fields of ANADIG Registers to the figure below.

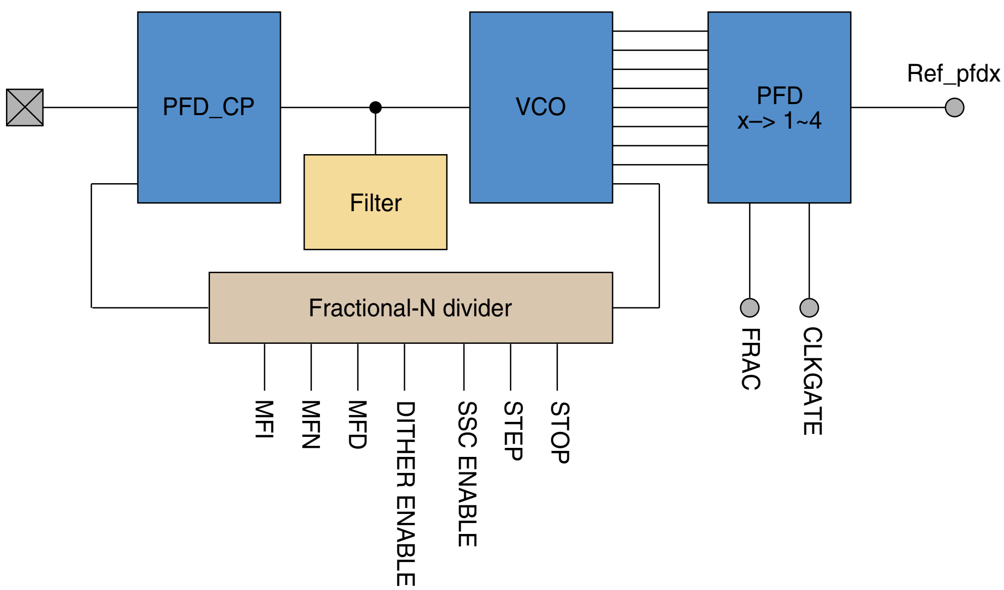


Figure 9-2. PLL Block Diagram

Primary Fractional Divider (PFD) — Each PFD works independently by interpolating the VCO of the PLL it is connected to. It takes the PLL VCO frequency and produces 18/N*Fvco its PFD output, where N ranges from 12 to 35. The time to switch frequencies is

much faster than a PLL, as the base PLL stays locked and changing the integer N just changes logical combination that controls the PFD output. These PFD outputs can be used by the system clocks or modules.

MFI	Controlled via DIV_SELECT field of ANADIG PLL Control registers
MFN	Controlled via ANADIG PLL2 Numerator definition register, ANADIG PLL1 Numerator definition register, ANADIG PLL4 Numerator definition register, ANADIG PLL6 Numerator definition register
MFD	Controlled via ANADIG PLL2 Denominator definition register, ANADIG PLL4 Denominator definition registerANADIG PLL6 Denominator definition register
STEP	Controlled via STEP field of ANADIG PLL2 Spread Spectrum definition register and PLL1 Spread Spectrum definition register
STOP	Controlled via STOP field of ANADIG PLL2 Spread Spectrum definition register and PLL1 Spread Spectrum definition register
SSC ENABLE	Controlled via ENABLE field of ANADIG PLL2 Spread Spectrum definition register and PLL1 Spread Spectrum definition register
DITHER ENABLE	Contfrolled via ANADIG PLL2 Control register, ANADIG PLL4 Control register, ANADIG PLL6 Control register, Anadig PLL5 Control register, and ANADIG PLL1 Control register
FRAC	Controlled via PFD1_FRAC, PFD2_FRAC, PFD3_FRAC, PFD3_FRAC, PFD4_FRAC fields of ANADIG_PLL3_PFD definition, ANADIG_PLL2_PFD, and ANADIG_PLL1_PFD register
CLKGATE	Controlled via PFD1_CLKGATE, PFD2_CLKGATE, PFD3_CLKGATE, PFD4_CLKGATE bits of ANADIG_PLL3_PFD definition, ANADIG_PLL2_PFD, and ANADIG_PLL1_PFD register

9.4.2 Spread Spectrum (SSC)

PLL1 and PLL2 support the Spread Spectrum (SSC) on its output. In SSC mode, the recommended modulation frequency is 30 KHz and 2% (peak-peak) frequency spread is supported. Using the equations below, the appropriate values of STOP and STEP can be derived. It is important to note that only Down spread frequency modulation is supported.

$$\text{Spread Spectrum Range} = F_{ref} * \left(\frac{STOP}{MFD}\right)$$

$$\text{Modulation Frequency} = F_{ref} * \left(\frac{STEP}{2*STOP}\right)$$

NOTE

"Spread Spectrum Range" is in Hz.

Example: To add 2% spread at 30kHz on default PLL1, the PLL1 MFD could be set to 10000 (this would not affect output as MFN = 0). The spread spectrum range would be 2% of 528MHz. From this we can work out that STOP should be set to 4400.

Using the recommended modulation frequency of 30kHz, the STEP value can be worked out to be 11.

F0 = Fref * N Where, N = Total Loop Division (Integer + Fraction)

PLL output frequency variation with a 2% modulation is shown in the figure below.

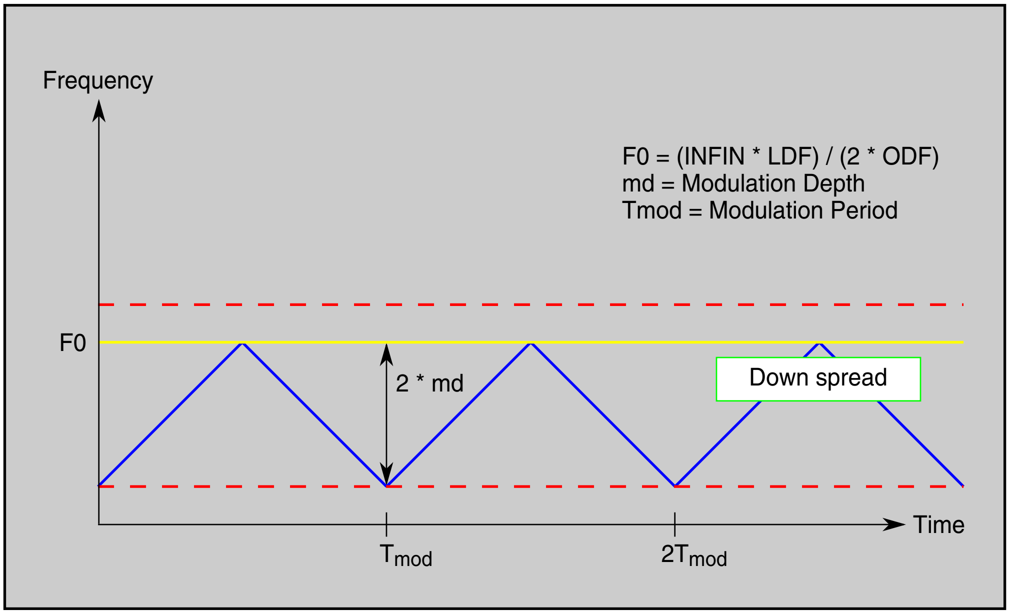


Figure 9-3. PLL with 2% Frequency Down spread Modulation

9.4.3 PLL Summary

The table below describes the properties of PLLs on this device.

Table 9-2. PLL Properties

		Basic			Spread Spectrum Support	PFD Output	Comments
PLL	**Description**	**Control**	**MFD**	**MFN**	**Available**	**Available**	**Comments**
PLL1 (PLLSYS 528 MHz)	This is a system PLL that generates the ARM subsystem clock.	Configurable DITHER and MFI	Configurable	Configurable	Yes	4	-
PLL2 (PLL 528 MHz)	This generates the PLL for DDR operation, as well as other modules.	Configurable DITHER and MFI	Configurable	Configurable	Yes	4	
PLL3 (USB0 PLL)	This generates a 480MHz clock to be used for USB0 and other modules.	Configurable DITHER. Fixed MFI.	NA (Fixed)	NA (Fixed)	NO	4	VCO output is fixed at 480 MHz, with MFI set to 20
PLL4 (Audio PLL)	This is a fractional multiplier PLL for generating Audio frequencies.	Configurable DITHER and MFI	Configurable	Configurable	NO	None	
PLL5 (ENET PLL)[1]	This PLL is used to generate 50 or 25 MHz for the external Ethernet interface.	Configurable DITHER and MFI	NA (Fixed)	NA (Fixed)	NO	None	VCO Output fixed at 1000 MHz
PLL6 (Video PLL)	This is a fractional multiplier PLL for generating Video frequencies.	Configurable DITHER and MFI	Configurable	Configurable	NO	None	

Table continues on the next page...

Table 9-2. PLL Properties (continued)

		Basic			Spread Spectrum Support	PFD Output	Comments
PLL	Description	Control	MFD	MFN	Available	Available	Comments
PLL 7 (USB1 PLL)	This generates a 480 MHz clock to be used for USB1.	No DITHER. Fixed MFI.	NA (Fixed)	NA (Fixed)	NO	None	

1. USB1 PHY has a separate internal PLL, PLL7 generating 480 MHz, from the 24 MHz FXOSC. This PLL is only used by USB1 PHY and does not have any PFDs

9.5 PLL Features

9.5.1 528 MHz Phase Locked Loop

Features:

- Synthesizes low jitter clock from 24MHz reference clock.
- Nine phase outputs of 528 MHz, plus 4 PFDs.
- Spread Spectrum support
- 2.5V ±10% Analog supply. 1.1V ±10% digital supply

$$\text{PLL Output Frequency} = F_{\text{ref}} * \left(\text{MFI} + \frac{\text{MFN}}{\text{MFD}}\right)$$

In the above equation:

- MFI - Multiplication Factor Integer (ANADIG_PLLx_CTRL[DIV_SELECT])
- MFN - Multiplication Factor Numerator (ANADIG_PLLx_NUM)
- MFD - Multiplication Factor Denominator (ANADIG_PLLx_DENOM)

$$\text{Spread Spectrum Range} = F_{\text{ref}} * \left(\frac{\text{STOP}}{\text{MFD}}\right)$$

$$\text{Modulation Frequency} = F_{\text{ref}} * \left(\frac{\text{STEP}}{2*\text{STOP}}\right)$$

$$\text{PFDn Output Frequency} = \text{PLLx Output Frequency} * \left(\frac{18}{\text{PFDn\_FRAC}}\right)$$

MFD is in ANADIG_PLLx_DENOM STEP and STOP are in ANADIG_PLLx_SS PFDn_FRAC is in ANADIG_PLLx_PFD[PFDn_FRAC]

For the electrical specifications, refer to the device Data Sheet.

NOTE

Registers that contain the configuration fields for the variables in the equations above for the 528 MHz PLLs are described in the ANADIG Registers chapter in this manual.

9.5.2 High Frequency PLL

There are two instances of this PLL:

- PLL4 - Used for audio clock generation
- PLL6 - Used for video clock generation

Features:

- Synthesizes low jitter clock from 24 MHz reference clock
- Clock output frequency range is from 650 MHz to 1.3 GHz
- Fractional-N synthesizer
- 2.5 V+-10% Analog supply. 1.1 V+-10% digital supply

PLL Output frequency is calculated based on the below equations:

$$\text{PLL Output Frequency} = F_{\text{ref}} * \left(\text{MFI} + \frac{\text{MFN}}{\text{MFD}}\right)$$

In the above equation:

- MFI - Multiplication Factor Integer (ANADIG_PLLx_CTRL[DIV_SELECT])
- MFN - Multiplication Factor Numerator (ANADIG_PLLx_NUM)
- MFD - Multiplication Factor Denominator (ANADIG_PLLx_DENOM)

$$\text{Modulation Frequency} = F_{\text{ref}} * \left(\frac{\text{STEP}}{2*\text{STOP}}\right)$$

Recommended modulation frequency is 30KHz. STEP and STOP are set in the ANADIG_PLLx_SS register.

For the Audio/Video PLL electrical specifications, refer to the device Data Sheet.

NOTE

Registers that contain the configuration fields for the variables in the equations above for the for Audio/Video PLLs described in the Anadig Registers chapter in this manual.

9.5.3 Ethernet PLL

Features:

- Synthesizes low jitter clock from 24 MHz reference clock.

- VCO OSC frequency = 1000 MHz / 2
- Outputs:
  - Programmable divider - 25/50/100 (ENET reference)

NOTE

Registers that contain the configuration fields for the variables in the equations above for the Ethernet PLL are described in the ANADIG Registers chapter in this manual. Some variables are fixed and are not available in the registers.

For the electrical specifications, refer to the device Data Sheet.

$$\text{PLL Output Frequency} = F_{ref} * \left(MFI + \frac{MFN}{MFD}\right)$$

Where:

- MFI: Multiplication Factor Integer
- MFN: Multiplication Factor Numerator
- MFD: Multiplication Factor Denominator

9.6 PLL/PFD Configuration

9.6.1 PLL Configuration

9.6.1.1 Typical PLL Configuration

Table 9-3. Typical PLL Configurations

Name	MFI	MFN	MFD	PFD	PFD Setting	Output (MHz)
PLL1 (PLLSYS)	ANADIG_PLL1_CTRL[DIV_SELECT]=1	0x0	0x12	PFD1	‘d19 (0x13)	500
				PFD2	‘d21 (0x15)	452
				PFD3	‘d24 (0x18)	396
				PFD4	‘d18 (0x12)	528
PLL2 (PLL528)	ANADIG_PLL2_CTRL[DIV_SELECT]=1 (24*22=528 MHz)	0x0	0x12	PFD1	‘d19 (0x13)	500
				PFD2	‘d24 (0x18)	396
				PFD3	‘d28 (0x1C)	339
				PFD4	‘d23 (0x17)	413

Table continues on the next page...

Table 9-3. Typical PLL Configurations (continued)

Name	MFI	MFN	MFD	PFD	PFD Setting	Output (MHz)
PLL 3 (USB0 PLL)	ANADIG_PLL3_CTRL[DIV_SELECT]=0 (24*20=480 MHz)			PFD1	'd28 (0x1C)	Actual o/p 308 MHz
				PFD2	‘d26 (0x1A)	332 MHz
				PFD3	‘d29 (0x1D)/ ’d28 (0x1C)	298 MHz/ 309 MHz
				PFD4	'd27 (0x1B)	320 MHz
PLL4 (Audio PLL) (default)	0x31	0x04DD2F15	0x1FFFFFDB	PLL Output = 24*49.152=1179.648 MHz **NOTE:** This when divided by 48 generates 512x48KHz		
PLL4 (other use case)	0x2F	0x0147AE13	0x1FFFFFDB	PLL output = 24*47.02=1128.48 MHz **NOTE:** This when divided by 50 generates 512x44.1KHz		
PLL5 (ENET PLL)	0x29 (tied internally)	0xAAAAD44 (tied internally)	0x100003E6 (tied internally)	VCO Clock = 24*41.66=1000, internally divided by 2 generates 500 MHz, **NOTE:** ANADIG_PLL5_CTRL[DIV_SELECT]=0x1 generates 50 MHz clock output.		
PLL6 (Video PLL)	0x2C	0xAAAAA9E	0x1FFFFFDA	PLL output = 24M*44.33=1064 MHz **NOTE:** This when divided by 8 generates 133 MHz		
PLL6 (default)	0x28	0x0	0x12	PLL output = 24*40=960 MHz **NOTE:** This when divided by 8 generates 120 MHz		
PLL7 (USB1 PLL)	ANADIG_PLL7_CTRL[DIV_SELECT]=0 (24*20=480 MHz)					

9.6.2 PFD Configuration

9.6.2.1 Typical PFD Configuration

Table 9-4. Typical PFD Configuration

PLL	PFD	Frequency	Notes
PLL1	PFD1	500	-
	PFD2	452	For Cortex-A5 clock
	PFD3	396	Synchronous Cortex-A5
	PFD4	528	To generate x4 clock of 264 MHz for QSPI in DDR66 mode

Table continues on the next page...

Table 9-4. Typical PFD Configuration (continued)

PLL	PFD	Frequency	Notes
PLL2	PFD1	500	Reserved
	PFD2	396	For DDR operation
	PFD3	339	For 339:166 synchronous DDR operation
	PFD4	413	For QSPI SDR 104 clock. Can also be used for GCC
PLL3	PFD1	308	Reserved for Audio clocking
	PFD2	332	-
	PFD3	298	Reserved for SDHC clocking. In case the above PFD1 is consumed in audio then we switch to PFD3
	PFD4	320	For QSPIx4 for DDR80. 320 MHz clk source requires setting of PFD4_Frac field in ANADIG_PLL3_PFD to 0x1b to allow DDR80 QuadSPI operation.

9.7 Clock Configuration

The table below provides the maximum frequencies of the listed peripherals on this device.

Table 9-5. Clocking Configuration

Clocks	Maximum Frequency
Cortex-A5 Core Clock[1]	500 MHz
Cortex-A5 Core Clock	166 MHz
DDR Clock[2]	400 MHz[3]
Platform Bus or Cortex-M4 Clock	166 MHz
IPS or IPG Clock. This is derived from (Platform Bus or Cortex-M4 Clock)/CCM_CACRR[IPG_CLK_DIV]. Typical value for this divider is 2.	83 MHz
DCU Pixel Clock	from 5 - 60 MHz
SDHC Clock	50 MHz
Video ADC (V_CLK1)	133 MHz
Ethernet (RMII_CLK)	50 MHz
Audio	24.58 MHz

1. Cortex-A5 core clock has to be synchronous to rest of the system.
2. DDR can be synchronous or asynchronous with a maximum speed of 400MHz.
3. Minimum frequency of DDR is 300 MHz

NOTE

On this device, the modules listed below have the register interface working at the same frequency as Platform Bus clock.

- DCU0
- DCU1
- DMA_CH_MUX0
- DMA_CH_MUX1
- DMA_CH_MUX2
- DMA_CH_MUX3
- RLE
- SAI0
- SAI1
- SAI2
- SAI3
- SPDIF
- GCC
- TCON

For other modules on this device, the register interface work at the IPS clock frequency, which is half of Platform Bus clock. For example:

- DAC
- PDB
- PIT
- LPTMR
- SPI
- I2C
- UART
- LCD
- VIU

9.8 Clock Modes

9.8.1 Synchronous Mode

9.8.1.1 Synchronous Mode

In this mode, Cortex-A5, BUS, Cortex-M4, and DDR clocks are synchronous to each other.

In the synchronous mode, all clocks are on the same source and have an integral ratio.

Table 9-6. Synchronous Mode

PFD Selected (Frequency)	Cortex-A5	DDR	BUS/Cortex-M4	IPG	Div Configurations
PLL1->PFD3 or PLL2->PFD2 (396MHz)	396	396	132	66	ARM_DIV = 1 BUS_DIV = 3 IPG_DIV = 2
PLL2->PFD3 (339Mhz)	339	339	169.5	84.75	ARM_DIV = 1 BUS_DIV = 2 IPG_DIV = 2
PLL1->PFD4 (528Mhz)	264	264	132	66	ARM_DIV = 2 BUS_DIV = 2 IPG_DIV = 2
PLL1->PFD1 or PLL2->PFD2 (500Mhz)	250	250	125	62.5	ARM_DIV = 2 BUS_DIV = 2 IPG_DIV = 2

9.8.2 Asynchronous Mode

9.8.2.1 Asynchronous DDR mode

In this mode, Cortex-A5, BUS, Cortex-M4 are synchronous and DDR clock is asynchronous.

In the asynchronous mode, DDR can be operated on different clock source for 400 MHz and Cortex-A5 can go up to 500 MHz. To set up the DDR asynchronous mode, DDR register configuration needs to be done. Refer to DDR Control Registers 117 .

Table 9-7. Asynchronous DDR mode

PFD Used	Cortex-A5	DDR	BUS/Cortex-M4	IPG	Div Configurations
PLL2->PFD3 (339Mhz)	339	396	169.5	84.75	• ARM_DIV = 1 • BUS_DIV = 2 • IPG_DIV = 2
PLL1->PFD2 (452Mhz)	452	396	150.66	75.33	• ARM_DIV = 1 • BUS_DIV = 3 • IPG_DIV =2

9.9 Clock Gating

9.9.1 Clock Gating

The device implements clock gating options for multiple peripherals to save power. The central clock gating is controlled through configuration of CCM Clock Gating Register (CCM_CCGCR) registers for off-platform modules and CCM Platform Clock Gating Register (CCM_CPGCR) for on platform modules. See the CCM chapter for details.

- All the on-platform module clocks are enabled by default.
- All the off-platform module clocks are disabled by default.
- Only CCM and SRC can be accessed without any IPS configuration.
- All auxillary clocks are implemented using asynchronous clock muxes. To avoid any glitch at the output, select the clock source and then enable the clock gating.

NOTE

A transfer error is observed if an access is made to a module that does not have a clock enabled.

NOTE

The Low Power Clock Gating module further gates the clock to various peripherals.

9.10 Peripheral Clocks

9.10.1 Module clocks

The following table summarizes the clocks associated with each module.

Table 9-8. Module clocks

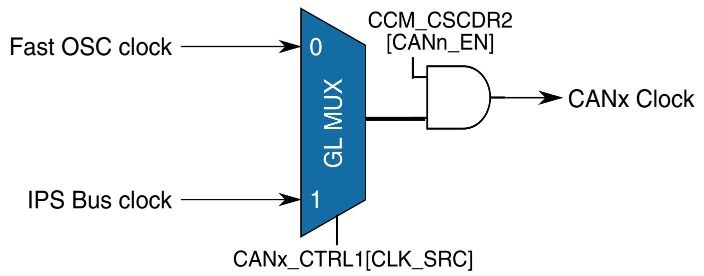
Modules	AHB	IPG Bus Clock	IPS Clock	Relation b/w IPG and IPS	RAM	Baud	Spl care on baud	Any other alternate	Comments
FlexRay	Yes	-	-	-	-	-	-	-	-
ENET_DPRAM	85	85	42.5	-	170	Auxiliary	mii_clk = 100, RMII = 50MHz	-	IPG clock > 50MHz to support both MII/RMII
DMA	SYS/AHB	-	-	-	85	No	-	-	-

Table continues on the next page...

Table 9-8. Module clocks (continued)

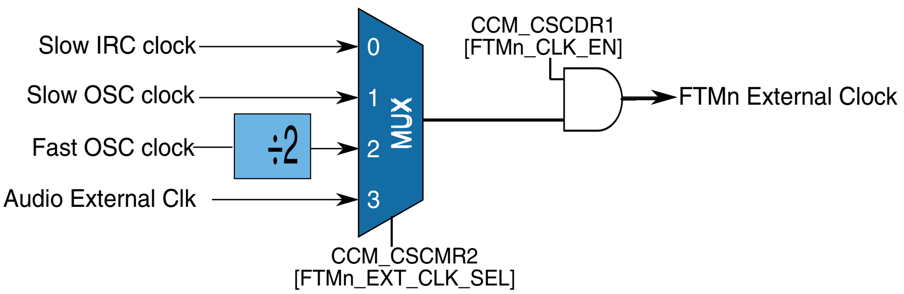
Modules	AHB	IPG Bus Clock	IPS Clock	Relation b/w IPG and IPS	RAM	Baud	Spl care on baud	Any other alternate	Comments
USB	85	-	-	-	85	-	-	-	-
SSCM	No	-	-	-	-	-	-	-	-
wkpu	No	-	-	-	-	-	-	-	-
I2C fil	No	-	-	-	-	-	-	-	-
CMU	No	-	-	-	-	-	-	-	-

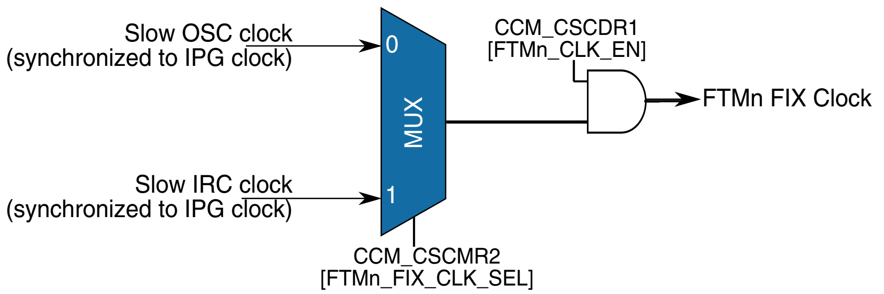
9.10.2 FlexCAN Clocking



9.10.3 FTM clocking

This section shows FlexTimer clocking.

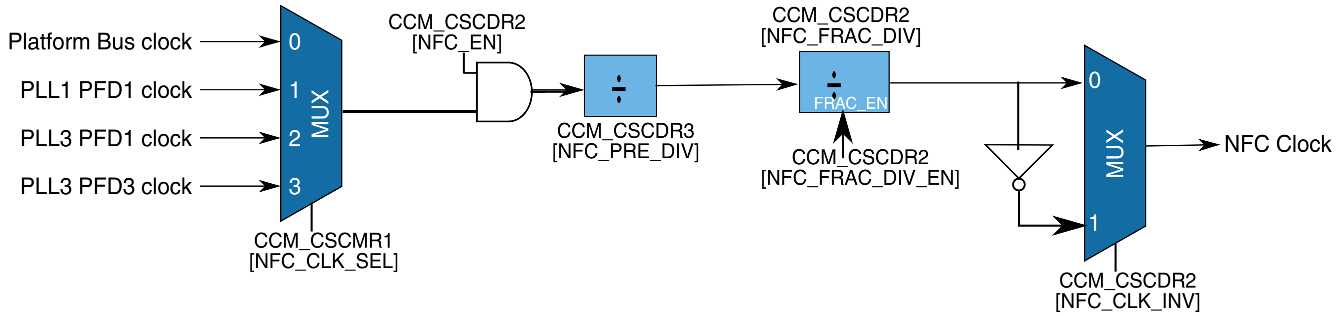




For the clock signal description, refer to the CCM chapter.

9.10.4 NFC clocking

This section shows NAND Flash Controller (NFC) clocking.



For the clock signal description, refer to the Clock Controller Module (CCM) chapter

9.10.5 QuadSPI Clocking

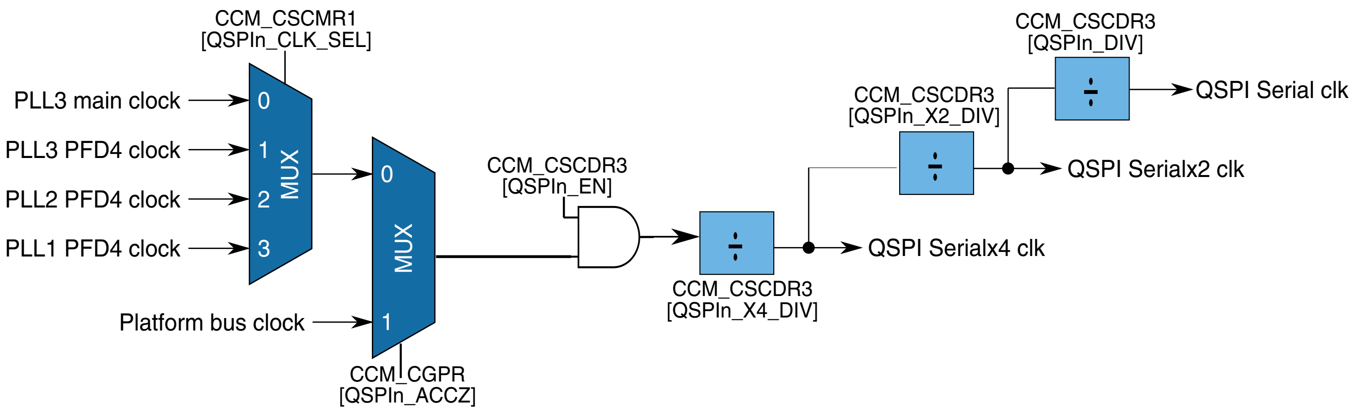
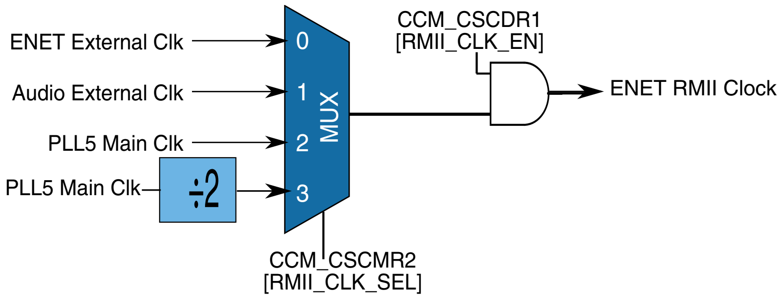


Figure 9-4. QuadSPI Clocking

Serial Flash Speed	QuadSPI Clock Requirements	PLL Option	Serialx4 Clk	Serialx2 Clk	Serial Clk
104 MHz SDR	104 MHz	ANADIG_PLL2_PFD[PFD4_FRAC]='d 23 (413MHz)		Not required	104 MHz
66 MHz DDR	132 MHz/264 MHz	ANADIG_PLL1_PFD[PFD4_FRAC]='d 18 (528MHz)	264 MHz	132 MHz	Yes
80 MHz DDR	160 MHz/320 MHz	ANADIG_PLL3_PFD[PFD4_FRAC]='d 27 (320MHz)	320 MHz	160 MHz	Yes

9.10.6 Ethernet RMII/MII Clocking

This section shows Ethernet RMII Clocking.

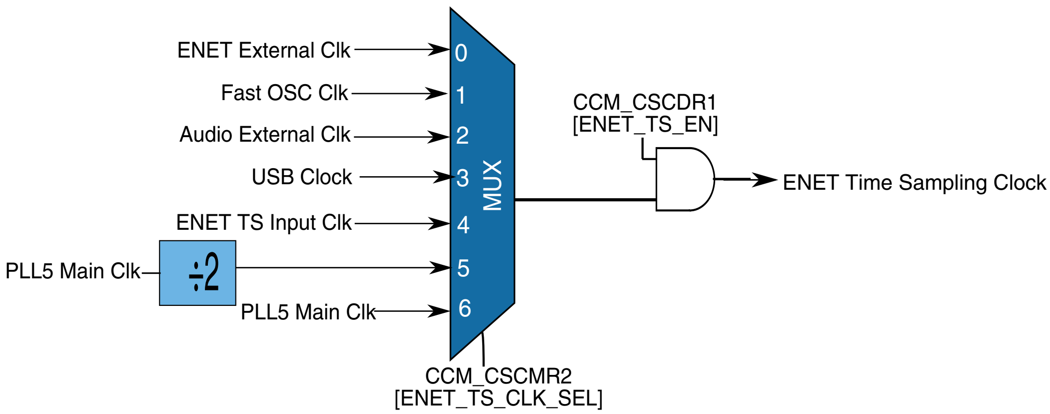


NOTE

When Ethernet MII clocking is used RMII clocking cannot be used. Ethernet MII TXClk is driven by ENET External Clk (PTA6/PTA9) and Ethernet MII RXClk is driven by PTA21.

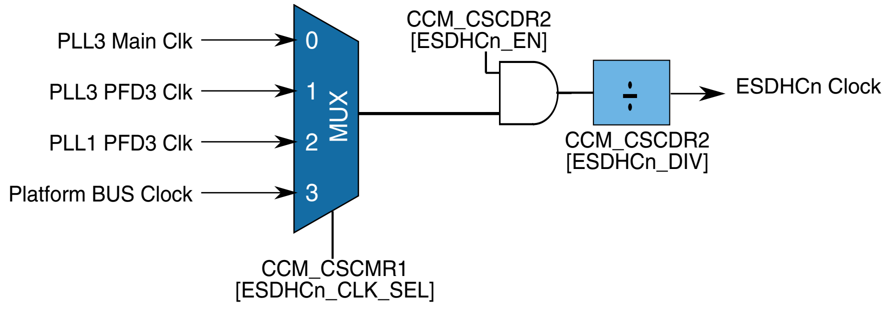
9.10.7 Ethernet Timer Clocking

This section shows Ethernet Timer clocking.



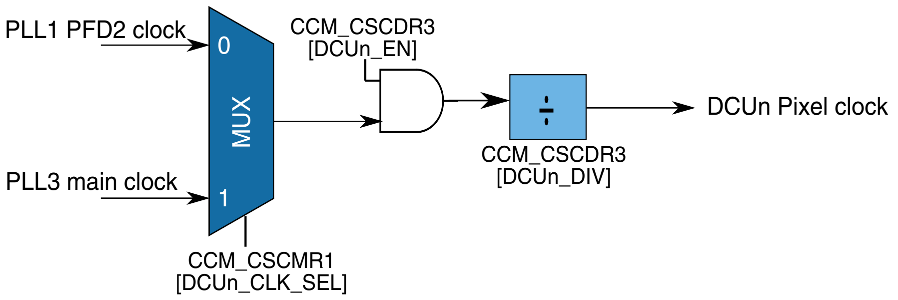
9.10.8 eSDHC Clocking

This section shows eSDHC clocking.



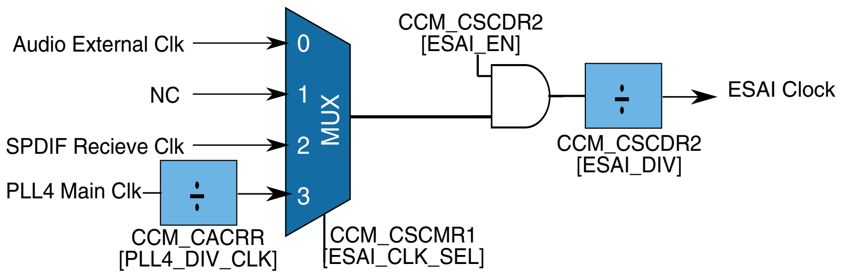
9.10.9 DCU clocking

This section shows DCU clocking.



9.10.10 ESAI clocking

This section shows ESAI clocking.

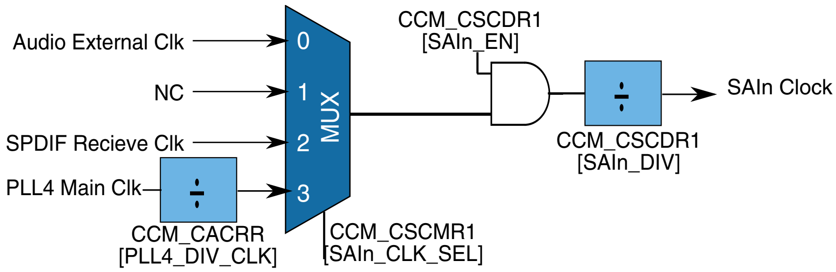


9.10.11 SPDIF Clocking

SPDIF clocking is controlled from the SPDIF module itself, through SPDIF_STC[TxClk_Source]. CCM has no direct control of SPDIF clock. SPDIF clock can be observed through CCM Clock Output Source Register (CCM_CCOSR).

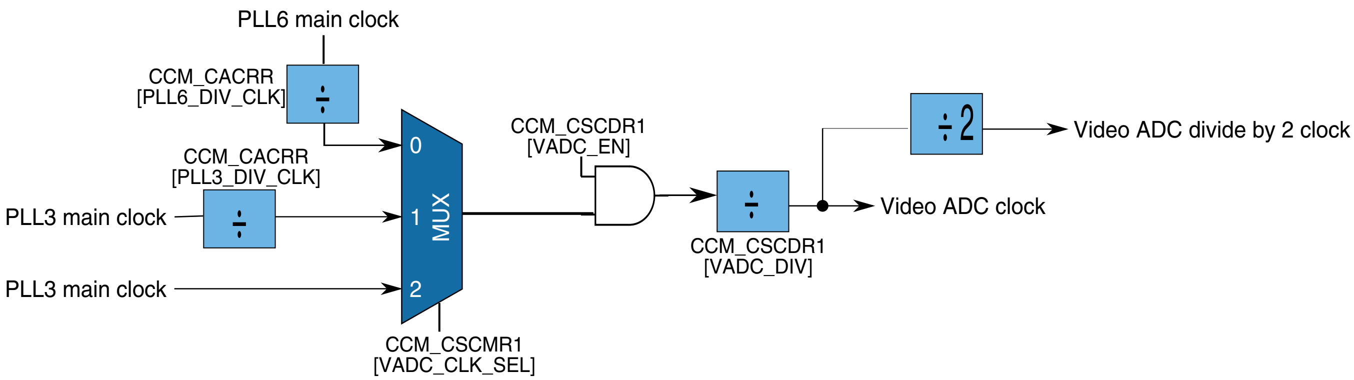
9.10.12 SAI clocking

This section shows SAI clocking.

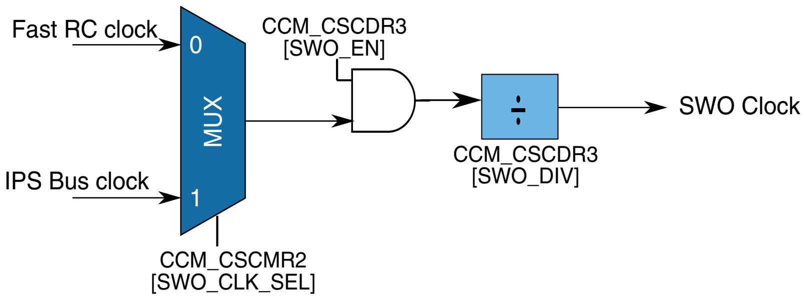


9.10.13 Video ADC clock

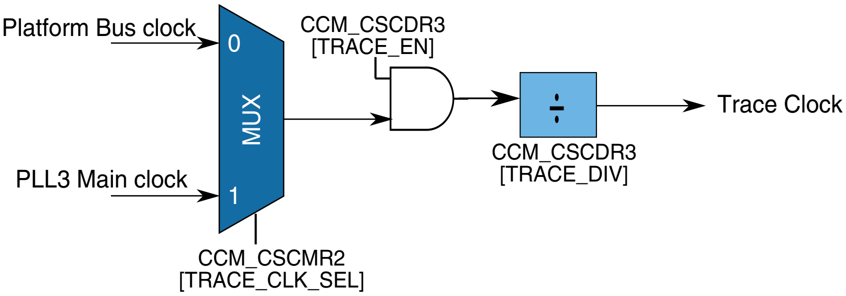
This section shows Video ADC clock.



9.10.14 SWO Clocking



9.10.15 Trace clocking



9.11 Appendix

9.12 Maximum Frequencies Supported

NOTE

Do not configure CCM to generate frequencies higher than those listed n the table below.

Table 9-10. Maximum frequencies

Clock name	Output/Input	Frequency (MHz)	Notes
Platform Bus Clock	output	170	
Cortex-A5 Clock	output	500	
FlexCAN Clock	output	85	
Cortex-M4 Clock	output	170	
DAP Clock	output	170	
DCU Pixel Clock	output	240	When pix_div is to be confirmed at 45/90, CCM should give out 90 MHz
DDRC Clock	output	400	
ENET Clock	input	50	
ENET RMII Clock	output	50	PLL5
ENET MII TX/RX Clock	input	25	
ENET TS Clock	input	100	
ENET Clock	output	50	
eSDHC Clock	output	52	PLL3_PFD3 (Divider inside CCM at 480 MHz)
flex_root_clk	output	85	
1588 TIMER Clock	output	100	PLL5

Table continues on the next page...

Table 9-10. Maximum frequencies (continued)

Clock name	Output/Input	Frequency (MHz)	Notes
IPG	output	85	
NFC Clock	output	80	
PLL1 Main Clock	input	528	
PLL1 PFD1 Clock	input	500	NFC Flash40
PLL1 PFD2 Clock	input	452	
PLL1 PFD3 Clock	input	396	
PLL1 PFD4 Clock	input	528	
PLL2 Main Clock	input	528	
PLL2 PFD1 Clock	input	500	
PLL2 PFD2 Clock	input	396	
PLL2 PFD3 Clock	input	339	
PLL2 PFD4 Clock	input	413	
PLL3 Main Clock	input	480	
PLL3 PFD3 Clock	input	308	
PLL3 PFD4 Clock	input	320	
PLL4 Main Clock	input	1300	
PLL5 Main Clock	input	100	
PLL5 Main Clock	input	1064	
QSPI Serial Clock	output	104	
QSPI SerialX2 Clock	output	160	
QSPI SerialX4 Clock	output	320	
SWO Clock	output	44	
Trace Clock	output	170	
VADC Core Clock	output	133	PLL6 : 1064/8
VADC Div Clock	output	66	

NOTE

Maximum baud rate supported at SPI output (serial) is 83 MHz.

Chapter 10
Clock Controller Module (CCM)

10.1 Introduction

This document describes the Clock Controller Module (CCM). The CCM generates the clocks for all the peripherals.

10.1.1 Overview

The Clock Controller Module controls the following functions:

- Uses the available clock sources to generate clock roots to various parts of the device
- Uses programmable bits to control frequencies of the clock roots
- Controls the low power mechanism
- Provides control signals to Low Power Clock Gating module (LPCG) for gating clocks
- Provides handshake with System Reset Controller (SRC) for reset performance
- Provides handshake with Global Power Controller (GPC) for low power mode operations

10.1.2 Features

The CCM includes the following features:

- Four sources of clocks: two internal RC oscillators and two crystal oscillators.
- Seven PLLs present in the processor.

PLL	PLL Defintion
PLL1	PLLSYS 528
PLL2	PLL 528

Table continues on the next page...

PLL	PLL Defintion
PLL3	USB0 PLL 480
PLL4	AUDIO PLL
PLL5	ENET PLL
PLL6	VIDEO PLL
PLL7	USB1 PLL

- Separate dividers and clock source selectors for core, bus, and each peripheral's clock.
- External clocks have option to bypass PLL clocks.
- Clock signals can be output on CKO1 and CKO2 pins for observation.
- Registers accessible via IP bus.
- Low Power modes management.
- Programmable clock gating of the peripheral clocks in low power modes.
- Frequency scaling management procedure for ARM core clock by shifting between PLL sources, without loss of clocks.
- Frequency scaling management procedure for peripheral root clock by programmable divider. The division is done on the fly without loss of clocks.

10.1.3 CCM BLOCK DIAGRAM

The CCM contains the following sub-blocks:

Table 10-1. CCM Sub-blocks

CCM_CLK_IGNITION	Manages the ignition process. This module starts its functionality once CCM comes out of reset. It manages the process that begins with starting the osc, PLLs and finishes with creation of stable output root clocks after reset.
CCM_CLK_SRC_DIV	Muxes different clocks and critical internal signals for observability. These output clocks are connected to the pads.
CCM_CLK_ROOT_GEN	Receives the main clocks and generates the output root clocks.
CCM_CLK_MOD_GEN	Generates the accessory clocks.
CCM_REG_IPS	The Register interface for configuring CCM and maintaining status.
CCM_LPM	Manages the low power mode entry and exit sequence.
CCM_CLK_LOGIC	Generates the clock enable signals based on info from CCM_LPM and CCM registers. The clock enables are used in LPCG to turn off and on the module clocks.

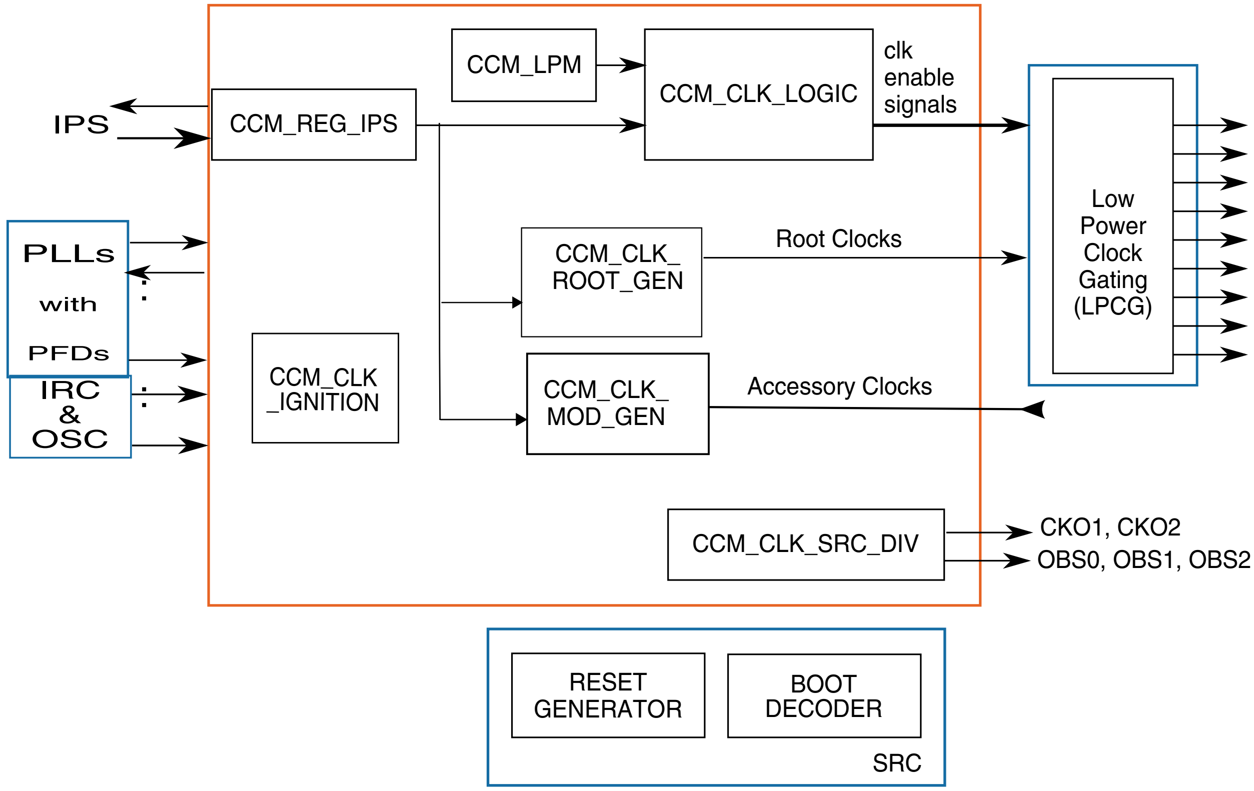


Figure 10-1. Block Diagram

10.2 Memory Map and Registers

CCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_B000	CCM Control Register (CCM_CCR)	32	R/W	0001_1057h	10.2.1/635
4006_B004	CCM Status Register (CCM_CSR)	32	R	0000_0000h	10.2.2/637
4006_B008	CCM Clock Switcher Register (CCM_CCSR)	32	R/W	0000_0000h	10.2.3/638
4006_B00C	CCM ARM Clock Root Register (CCM_CACRR)	32	R/W	0060_0008h	10.2.4/640
4006_B010	CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1)	32	R/W	0000_0000h	10.2.5/643
4006_B014	CCM Serial Clock Divider Register 1 (CCM_CSCDR1)	32	R/W	0000_0000h	10.2.6/645
4006_B018	CCM Serial Clock Divider Register 2 (CCM_CSCDR2)	32	R/W	0000_0000h	10.2.7/647
4006_B01C	CCM Serial Clock Divider Register 3 (CCM_CSCDR3)	32	R/W	0000_0000h	10.2.8/650
4006_B020	CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2)	32	R/W	0000_0000h	10.2.9/652
4006_B028	CCM Testing Observability Register (CCM_CTOR)	32	R/W	0000_0000h	10.2.10/ 655

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_B02C	CCM Low Power Control Register (CCM_CLPCR)	32	R/W	0000_0878h	10.2.11/ 657
4006_B030	CCM Interrupt Status Register (CCM_CISR)	32	w1c	0000_0000h	10.2.12/ 661
4006_B034	CCM Interrupt Mask Register (CCM_CIMR)	32	R/W	FFFF_FFFFh	10.2.13/ 663
4006_B038	CCM Clock Output Source Register (CCM_CCOSR)	32	R/W	000A_0001h	10.2.14/ 664
4006_B03C	CCM General Purpose Register (CCM_CGPR)	32	R/W	0000_0000h	10.2.15/ 668
4006_B040	CCM Clock Gating Register (CCM_CCGR0)	32	R/W	0000_0000h	10.2.16/ 669
4006_B044	CCM Clock Gating Register (CCM_CCGR1)	32	R/W	0000_0000h	10.2.16/ 669
4006_B048	CCM Clock Gating Register (CCM_CCGR2)	32	R/W	0000_0000h	10.2.16/ 669
4006_B04C	CCM Clock Gating Register (CCM_CCGR3)	32	R/W	0000_0000h	10.2.16/ 669
4006_B050	CCM Clock Gating Register (CCM_CCGR4)	32	R/W	0000_0000h	10.2.16/ 669
4006_B054	CCM Clock Gating Register (CCM_CCGR5)	32	R/W	0000_0000h	10.2.16/ 669
4006_B058	CCM Clock Gating Register (CCM_CCGR6)	32	R/W	0000_0000h	10.2.16/ 669
4006_B05C	CCM Clock Gating Register (CCM_CCGR7)	32	R/W	0000_0000h	10.2.16/ 669
4006_B060	CCM Clock Gating Register (CCM_CCGR8)	32	R/W	0000_0000h	10.2.16/ 669
4006_B064	CCM Clock Gating Register (CCM_CCGR9)	32	R/W	0000_0000h	10.2.16/ 669
4006_B068	CCM Clock Gating Register (CCM_CCGR10)	32	R/W	0000_0000h	10.2.16/ 669
4006_B06C	CCM Clock Gating Register (CCM_CCGR11)	32	R/W	0000_0000h	10.2.16/ 669
4006_B070	CCM Module Enable Override Register (CCM_CMEOR0)	32	R/W	0000_0000h	10.2.17/ 676
4006_B074	CCM Module Enable Override Register (CCM_CMEOR1)	32	R/W	0000_0000h	10.2.17/ 676
4006_B078	CCM Module Enable Override Register (CCM_CMEOR2)	32	R/W	0000_0000h	10.2.17/ 676
4006_B07C	CCM Module Enable Override Register (CCM_CMEOR3)	32	R/W	0000_0000h	10.2.17/ 676
4006_B080	CCM Module Enable Override Register (CCM_CMEOR4)	32	R/W	0000_0000h	10.2.17/ 676

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_B084	CCM Module Enable Override Register (CCM_CMEOR5)	32	R/W	0000_0000h	10.2.17/ 676
4006_B088	CCM PLL PFD Disable Status Register (CCM_CPPDSR)	32	R	0000_0FFFh	10.2.18/ 679
4006_B08C	CCM CORE Wakeup Register (CCM_CCOWR)	32	R/W	0000_0000h	10.2.19/ 681
4006_B090	CCM Platform Clock Gating Register (CCM_CCPGR0)	32	R/W	See section	10.2.20/ 682
4006_B094	CCM Platform Clock Gating Register (CCM_CCPGR1)	32	R/W	See section	10.2.20/ 682
4006_B098	CCM Platform Clock Gating Register (CCM_CCPGR2)	32	R/W	See section	10.2.20/ 682
4006_B09C	CCM Platform Clock Gating Register (CCM_CCPGR3)	32	R/W	See section	10.2.20/ 682

10.2.1 CCM Control Register (CCM_CCR)

The figure below represents the CCM Control Register (CCR), which contains bits to control general operation of CCM. The table below provides its field descriptions.

Address: 4006_B000h base + 0h offset = 4006_B000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															FIRC_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			FXOSC_EN	0				OSCNT							
W																
Reset	0	0	0	1	0	0	0	0	0	1	0	1	0	1	1	1

CCM_CCR field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

CCM_CCR field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>16
FIRC_EN</td><td>On-Chip fast RC Oscillator enable bit

0 Disable on-chip RC oscillator
1 Enable on-chip RC oscillator</td></tr>
<tr><td>15–13
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>12
FXOSC_EN</td><td>On-chip fast crystal oscillator (24 MHz FXOSC) enable bit. The system will start with on-chip oscillator enabled to supply source for the PLL’s. Software can change this bit if a transition to the bypass PLL clocks was performed for all the PLLs. In cases that this bit is changed from ‘0’ to ‘1’ then CCM will enable the on-chip oscillator and after counting CCM_CCR[OSCNT] clock cycles it will notify that on-chip oscillator is ready by a interrupt FXOSC_RDY and by status bit FXOSC_RDY. The FXOSC_EN bit should be changed only when on-chip oscillator is not chosen as the clock source.

Table 10-4. CCR[FXOSC_EN] and CLPCR[FXOSC_PWRDWN] behavior
<table>
<tr><th>FXOSC_EN</th><th>FXOSC_PWRDWN</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>FXOSC clock is enabled but FXOSC_OK is disabled.</td></tr>
<tr><td>1</td><td>0</td><td>FXOSC clock and FXOSC_OK are enabled.</td></tr>
<tr><td>0</td><td>1</td><td>FXOSC is power down and clock is not coming. FXOSC_OK is disabled.</td></tr>
<tr><td>1</td><td>1</td><td>FXOSC is power down and clock is not coming. Also FXOSC_OK is disabled.</td></tr>
</table>
</td></tr>
<tr><td>11–8
Reserved</td><td>This field is reserved.
This read-only field is reserved and always has the value 0.</td></tr>
<tr><td>7–0
OSCNT</td><td>Oscillator ready counter value. These bits define value of 32 KHz counter, that serve as counter for oscillator lock time. This is used for oscillator lock time. Current estimation is ~2.7ms. This counter will be used in ignition sequence and in wake from stop sequence if CCM_CLPCR[SBYOS] bit was defined, to notify that on-chip oscillator output is ready for use.

00000000 count 1 cycle of 32 KHz SXOSC clock
11111111 count 256 cycles of 32 KHz SXOSC clock</td></tr>
</table>

10.2.2 CCM Status Register (CCM_CSR)

The figure below represents the CCM status Register (CSR). The status bits are read only bits. The table below provides its field descriptions.

Address: 4006_B000h base + 4h offset = 4006_B004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										FXOSC_RDY	1	0			Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CSR field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 FXOSC_RDY	Status indication of on board oscillator. This bit will be asserted after CCM_CCR[OSCNT] cycles after the on-chip oscillator is powered on and enabled. 0 On board oscillator is not ready. 1 On board oscillator is ready.

Table continues on the next page...

CCM_CSR field descriptions (continued)

Field	Description
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	This field is reserved. This bit is reserved. Writes to this bit have no effect.

10.2.3 CCM Clock Switcher Register (CCM_CCSR)

The figure below represents the CCM Clock Switcher register (CCSR). The CCSR register contains bits to control the switcher sub module dividers and multiplexers. The table below provides its field descriptions.

Address: 4006_B000h base + 8h offset = 4006_B008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PLL3_PFD4_EN	PLL3_PFD3_EN	PLL3_PFD2_EN	PLL3_PFD1_EN	0			DAP_EN	0		PLL2_PFD_CLK_SEL			PLL1_PFD_CLK_SEL		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PLL2_PFD4_EN	PLL2_PFD3_EN	PLL2_PFD2_EN	PLL2_PFD1_EN	PLL1_PFD4_EN	PLL1_PFD3_EN	PLL1_PFD2_EN	PLL1_PFD1_EN	0	DDRC_CLK_SEL	FAST_CLK_SEL	SLOW_CLK_SEL	Reserved	SYS_CLK_SEL		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CCSR field descriptions

Field	Description
31 PLL3_PFD4_EN	Enable for PLL3 PFD4 0 Disable PLL3 PFD4 (will override the CCM internally generated enables) 1 Enable PLL3 PFD4 (PLL PFD may still be disabled, if not used by CCM)
30 PLL3_PFD3_EN	Enable for PLL3 PFD3 0 Disable PL3 PFD3 (will override the CCM internally generated enables) 1 Enable PLL3 PFD3 (PLL PFD may still be disabled, if not used by CCM)

Table continues on the next page...

CCM_CCSR field descriptions (continued)

Field	Description
29 PLL3_PFD2_EN	Enable for PLL3 PFD2 0 Disable PLL3 PFD2 (will override the CCM internally generated enables) 1 Enable PLL3 PFD2 (PLL PFD may still be disabled, if not used by CCM)
28 PLL3_PFD1_EN	Enable for PLL3 PFD1 0 Disable PLL3 PFD1 (will override the CCM internally generated enables) 1 Enable PLL3 PFD1 (PLL PFD may still be disabled, if not used by CCM)
27–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 DAP_EN	Enable for Debug Access port clock 0 Disable Debug Access Port clock 1 Enable Debug Access Port clock, an acknowledgement is sent to Debug Access port once the clock is enabled.
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–19 PLL2_PFD_CLK_SEL	PLL2 PFD clock select 000 PLL2 main clock 001 PLL2 PFD1 clock 010 PLL2 PFD2 clock 011 PLL2 PFD3 clock 100 PLL2 PFD4 clock
18–16 PLL1_PFD_CLK_SEL	PLL1 PFD clock select 000 PLL1 main clock 001 PLL1 PFD1 clock 010 PLL1 PFD2 clock 011 PLL1 PFD3 clock 100 PLL1 PFD4 clock
15 PLL2_PFD4_EN	Enable for PLL2 PFD4 0 Disable PLL2 PFD4 (will override the CCM internally generated enables) 1 Enable PLL2 PFD4 (PLL PFD may still be disabled, if not used by CCM)
14 PLL2_PFD3_EN	Enable for PLL2 PFD3 0 Disable PLL2 PFD3 (will override the CCM internally generated enables) 1 Enable PLL2 PFD3 (PLL PFD may still be disabled, if not used by CCM)
13 PLL2_PFD2_EN	Enable for PLL2 PFD2 0 PLL2 PFD2 (will override the CCM internally generated enables) 1 Enable PLL2 PFD2 (PLL PFD may still be disabled, if not used by CCM)
12 PLL2_PFD1_EN	Enable for PLL2 PFD1 0 Disable PLL2 PFD1 (will override the CCM internally generated enables) 1 Enable PLL2 PFD1 (PLL PFD may still be disabled, if not used by CCM)
11 PLL1_PFD4_EN	Enable for PLL1 PFD4

Table continues on the next page...

CCM_CCSR field descriptions (continued)

Field	Description
	0 Disable PLL1 PFD4 (will override the CCM internally generated enables) 1 Enable PLL1 PFD4 (PLL PFD may still be disabled, if not used by CCM)
10 PLL1_PFD3_EN	Enable for PLL1 PFD3 0 Disable PLL1 PFD3 (will override the CCM internally generated enables) 1 Enable PLL1 PFD3 (PLL PFD may still be disabled, if not used by CCM)
9 PLL1_PFD2_EN	Enable for PLL1 PFD2 0 Disable PLL1 PFD2 (will override the CCM internally generated enables) 1 Enable PLL1 PFD2 (PLL PFD may still be disabled, if not used by CCM)
8 PLL1_PFD1_EN	Enable for PLL1 PFD1 0 Disable PLL1 PFD1 (will override the CCM internally generated enables) 1 Enable PLL1 PFD1 (PLL PFD may still be disabled, if not used by CCM)
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 DDRC_CLK_SEL	DDRC clock select 0 PLL2 PFD2 clk 1 SYS_DIV_OUT_CLK
5 FAST_CLK_SEL	Fast clock select 0 24 MHz IRC clock 1 24 MHz FXOSC clock
4 SLOW_CLK_SEL	Slow clock select 0 32 KHz divided 128 kHz IRC clock 1 32 KHz FXOSC clock
3 Reserved	This field is reserved. This is read-write bit but it should not be written and always be 0.
2–0 SYS_CLK_SEL	System clock select 000 Fast clock o/p defined by CCM_CCSR[FAST_CLK_SEL] 001 Slow clock o/p defined by CCM_CCSR[SLOW_CLK_SEL] 010 PLL2 PFD o/p clock defined by CCM_CCSR[PLL2_PFD_CLK_SEL] 011 PLL2 main clock 100 PLL1 PFD o/p clock defined by CCM_CCSR[PLL1_PFD_CLK_SEL] 101 PLL3 main clock

10.2.4 CCM ARM Clock Root Register (CCM_CACRR)

The figure below represents the CCM ARM Clock Root register (CACRR). The CACRR register contains bits to control the ARM clock root generation. The table below provides its field descriptions.

NOTE

The divider value should be changed only when its not selected as the source.

Address: 4006_B000h base + Ch offset = 4006_B00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							FLEX_CLK_DIV			PLL6_CLK_DIV	PLL3_CLK_DIV	0		PLL1_PFD_CLK_DIV	
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			IPG_CLK_DIV		0		PLL4_CLK_DIV			BUS_CLK_DIV			ARM_CLK_DIV		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

CCM_CACRR field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24–22 FLEX_CLK_DIV	FLEX clock divider value 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
21 PLL6_CLK_DIV	PLL6 divider select (before switching the clocks should be gated) 0 Divide by 1 (used only if PLL is less than or equal to 650 MHz) 1 Divide by 2
20 PLL3_CLK_DIV	PLL3 divider select 0 Divide by 1 (default) 1 Divide by 2
19–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 PLL1_PFD_CLK_DIV	PLL1 PFD clock divider 00 Divide by 1 (default) 01 Divide by 2

Table continues on the next page...

CCM_CACRR field descriptions (continued)

Field	Description
	10 Divide by 3 11 Divide by 4
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. Reseved. Reads only zero
12–11 IPG_CLK_DIV	IP clock divider 00 Divide by 1 01 Divide by 2 10 Divide by 3 11 Divide by 4
10–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–6 PLL4_CLK_DIV	PLL4 clock divider (before switching the clocks should be gated) 000 Divide by 1 (only if PLL frequency less than or equal to 650 MHz) 001 Divide by 4 010 Divide by 6 011 Divide by 8 100 Divide by 10 101 Divide by 12 110 Divide by 14 111 Divide by 16
5–3 BUS_CLK_DIV	BUS clock divider 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
2–0 ARM_CLK_DIV	Divider for ARM clock root 000 Divide by 1(default) 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

10.2.5 CCM Serial Clock Multiplexer Register 1 (CCM_CSCMR1)

The figure below represents the CCM Serial Clock Multiplexer Register 1 (CSCMR1). The CSCMR1 register contains bits to control the multiplexers that generate the module's clocks. The table below provides its field descriptions.

NOTE

Any change on the clock selection of the module must be done when the clock of the module is gated.

Address: 4006_B000h base + 10h offset = 4006_B010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		DCU1_CLK_SEL	DCU0_CLK_SEL	0		QSPI1_CLK_SEL		QSPI0_CLK_SEL		ESAI_CLK_SEL		ESDHC1_CLK_SEL		ESDHC0_CLK_SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	Reserved	NFC_CLK_SEL		0		VADC_CLK_SEL		SAI3_CLK_SEL		SAI2_CLK_SEL		SAI1_CLK_SEL		SAI0_CLK_SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CSCMR1 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 DCU1_CLK_SEL	DCU1 PIX DIV clock select 0 PLL1 PFD2 clock 1 PLL3 MAIN clock
28 DCU0_CLK_SEL	DCU0 PIX DIV Clock Select 0 PLL1 PFD2 Clk 1 PLL3 MAIN Clk
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 QSPI1_CLK_SEL	Selector for QSPI2 clock multiplexer 00 PLL3 main clock 01 PLL3 PFD4

Table continues on the next page...

CCM_CSCMR1 field descriptions (continued)

Field	Description
	10 PLL2 PFD4 11 PLL1 PFD4
23–22 QSPI0_CLK_SEL	Selector for QSPI1 clock multiplexer 00 PLL3 main clock 01 PLL3 PFD4 10 PLL2 PFD4 11 PLL1 PFD4
21–20 ESAI_CLK_SEL	Selector for ESAI clock multiplexer 00 Audio External clock 01 Reserved 10 SPDIF RX Clk 11 Divided PLL4 main clock, defined by CCM_CACRR[PLL4_CLK_DIV]
19–18 ESDHC1_CLK_ SEL	ESDHC1 clock select 00 PLL3 main clock (default) 01 PLL3 PFD3 10 PLL1 PFD3 11 Platform bus clock
17–16 ESDHC0_CLK_ SEL	ESDHC0 clock select 00 PLL3 main clock (default) 01 PLL3 PFD3 10 PLL1 PFD3 11 Platform bus clock
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 Reserved	Reserved This field is reserved.
13–12 NFC_CLK_SEL	Selector for NAND Flash Controller clock multiplexer 00 Platform bus clock 01 PLL1 PFD1 clock 10 PLL3 PFD1 clock 11 PLL3 PFD3 clock
11–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 VADC_CLK_SEL	Video ADC clock select 00 Divided PLL6 main clock, defined by CCM_CACRR[PLL6_CLK_DIV] 01 Divided PLL3 main clock, defined by CCM_CACRR[PLL3_CLK_DIV] 10 PLL3 main clock 11 Reserved
7–6 SAI3_CLK_SEL	Selector for SAI3 clock multiplexer 00 Audio external clock

Table continues on the next page...

CCM_CSCMR1 field descriptions (continued)

Field	Description
	01 Reserved 10 SPDIF RX clock 11 Divided PLL4 main clock, defined by CCM_CACRR[PLL4_CLK_DIV]
5–4 SAI2_CLK_SEL	Selector for SAI2 clock multiplexer 00 Audio external clock 01 10 SPDIF RX Clk 11 Divided PLL4 Main clock, defined by CCM_CACRR[PLL4_CLK_DIV]
3–2 SAI1_CLK_SEL	Selector for SAI1 clock multiplexer 00 Audio external clock 01 10 SPDIF RX clock 11 Divided PLL4 main clock, defined by CCM_CACRR[PLL4_CLK_DIV]
1–0 SAI0_CLK_SEL	Selector for SAI0 clock multiplexer 00 Audio external clock 01 10 SPDIF RX clock 11 Divided PLL4 main clock, defined by CCM_CACRR[PLL4_CLK_DIV]

10.2.6 CCM Serial Clock Divider Register 1 (CCM_CSCDR1)

The figure below represents the CCM Serial Clock Divider Register 1 (CSCDR1). The CSCDR1 register contains bits to control the clock generation sub module dividers. The table below provides its field descriptions.

NOTE

Any change on the dividers should to be done while the module clock is gated.

Address: 4006_B000h base + 14h offset = 4006_B014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			FTM3_CLK_EN	FTM2_CLK_EN	FTM1_CLK_EN	FTM0_CLK_EN	RMII_CLK_EN	ENET_TS_EN	VADC_EN	VADC_DIV		SAI3_EN	SAI2_EN	SAI1_EN	SAI0_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	SAI3_DIV				SAI2_DIV				SAI1_DIV				SAI0_DIV			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CSCDR1 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 FTM3_CLK_EN	FTM3 clock enable 0 FTM clock disabled 1 FTM3 clock enabled
27 FTM2_CLK_EN	FTM2 clock enable 0 FTM2 clock disabled 1 FTM2 clock enabled
26 FTM1_CLK_EN	FTM1 clock enable 0 FTM1 clock disabled 1 FTM1 clock enabled
25 FTM0_CLK_EN	FTM 0 clock enable 0 FTM0 clock disabled 1 FTM1 clock enabled
24 RMII_CLK_EN	ENET RMII clock enable 0 Disable ENET RMII clock 1 Enable ENET RMII clock
23 ENET_TS_EN	ENET Time sampling clock enable 0 Disable ENET TS clock 1 Enable ENET TS clock
22 VADC_EN	Video ADC clock enable 0 Disable Video ADC clock 1 Enable Video ADC clock
21–20 VADC_DIV	Divider to generate Video ADC clock 00 Divide by 1 01 Divide by 2 10 Divide by 3 11 Divide by 4
19 SAI3_EN	SAI3 clock enable 0 Disable SAI3 clock 1 Enable SAI3 clock

Table continues on the next page...

CCM_CSCDR1 field descriptions (continued)

Field	Description
18 SAI2_EN	SAI2 clock enable 0 Disable SAI2 clock 1 Enable SAI2 clock
17 SAI1_EN	SAI1 clock enable 0 Disable SAI1 clock 1 Enable SAI1 clock
16 SAI0_EN	SAI0 clock enable 0 Disable SAI0 clock 1 Enable SAI0 clock
15–12 SAI3_DIV	Divider to generate SAI3 clock 0000 Divide by 1 0001 Divide by 2 ... ... 1111 Divide by 16
11–8 SAI2_DIV	Divider to generate SAI2 clock 0000 Divide by 1 0001 Divide by 2 ... ... 1111 Divide by 16
7–4 SAI1_DIV	Divider to generate SAI1 clock 0000 Divide by 1 0001 Divide by 2 ... ... 1111 Divide by 16
3–0 SAI0_DIV	Divider to generate SAI0 clock 0000 Divide by 1 0001 Divide by 2 ... ... 1111 Divide by 16

10.2.7 CCM Serial Clock Divider Register 2 (CCM_CSCDR2)

The figure below represents the CCM Serial Clock Divider Register 2(CSCDR2). The CSCDR2 register contains bits to control the clock generation sub module dividers. The table below provides its field descriptions.

NOTE

Any change on the dividers have to be done while the module clock is gated.

NOTE

When NFC_FRAC_DIV_EN is 1, do not configure NFC_FRAC_DIV[7:4] to 1111. The device does not support the divider value of 16.5.

Address: 4006_B000h base + 18h offset = 4006_B018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	ESAI_EN	ESDHC1_EN	ESDHC0_EN	ESAI_DIV[3:0]				ESDHC1_DIV[3:0]				ESDHC0_DIV[3:0]			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	NFC_CLK_INV	NFC_FRAC_DIV_EN	CAN1_EN	CAN0_EN	Reserved	NFC_EN	0	NFC_FRAC_DIV				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CSCDR2 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 ESAI_EN	ESAI clock enable 0 Disable ESAI clock 1 Enable ESAI clock
29 ESDHC1_EN	ESDHC1 clock enable 0 Disable ESDHC2 clock 1 Enable ESDHC2 clock
28 ESDHC0_EN	ESDHC0 clock enable 0 Disable ESDHC1 clock 1 Enable ESDHC1 clock
27–24 ESAI_DIV[3:0]	Divider to generate ESAI clock 0000 Divide by 1 0001 Divide by 2 ... ... 1111 Divide by 16
23–20 ESDHC1_ DIV[3:0]	Divider to generate ESDHC2 clock 0000 Divide by 1

Table continues on the next page...

CCM_CSCDR2 field descriptions (continued)

Field	Description
	0001 Divide by 2 ... ... 1111 Divide by 16
19–16 ESDHC0_ DIV[3:0]	Divider to generate ESDHC1 clock 0000 Divide by 1 0001 Divide by 2 ... ... 1111 Divide by 16
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 NFC_CLK_INV	NFC clock phase select 0 Low Phase of NFC clock > 50% 1 High Phase of NFC clock > 50%
13 NFC_FRAC_ DIV_EN	Enables the division in Fraction. If this bit is not set the NFC_FRAC_DIV divider acts as integer divider 0 NFC_FRAC_DIV divider acts as integer divider 1 Fractional 0.5 divider enabled (in addition to FRAC_DIV). Example: When NFC_FRAC_DIV = 1 and NFC_FRAC_DIV_EN =0 the division value is 1. When NFC_FRAC_DIV = 1 and NFC_FRAC_DIV_EN =1 the division value = 1.5.
12 CAN1_EN	CAN1 clock enable 0 Disable CAN1 clock 1 Enable CAN1 clock
11 CAN0_EN	CAN0 clock enable 0 Disable CAN0 clock 1 Enable CAN0 clock
10 Reserved	Reserved This field is reserved. Reserved
9 NFC_EN	NFC clock enable 0 Disable NFC clock 1 Enable NFC clock
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 NFC_FRAC_DIV	Divider to generate NAND Flash Controller clock 0000 Divide by 1 0001 Divide by 2 ... ... 1111 Divide by 16
3–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

10.2.8 CCM Serial Clock Divider Register 3 (CCM_CSCDR3)

The figure below represents the CCM Serial Clock Divider Register 3(CSCDR3). The CSCDR3 register contains bits to control the clock generation sub module dividers. The table below provides its field descriptions.

NOTE

Any change on the dividers have to be done while the module clock is gated.

Address: 4006_B000h base + 1Ch offset = 4006_B01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			SWO_EN	SWO_DIV	TRACE_EN	TRACE_DIV		DCU1_EN	DCU1_DIV			DCU0_EN	DCU0_DIV		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NFC_PRE_DIV			QSPI1_EN	QSPI1_DIV	QSPI1_X2_DIV	QSPI1_X4_DIV[1:0]		0			QSPI0_EN	QSPI0_DIV	QSPI0_X2_DIV	QSPI0_X4_DIV	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CSCDR3 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 SWO_EN	SWO clock enable 0 Disable SWO clock 1 Enable SWO clock
27 SWO_DIV	Divider to generate SWO clock 0 Divide by 1 1 Divide by 2
26 TRACE_EN	Trace clock enable 0 Disable Trace clock 1 Enable Trace clock
25–24 TRACE_DIV	Divider to generate Trace clock 00 Divide by 1 01 Divide by 2

Table continues on the next page...

CCM_CSCDR3 field descriptions (continued)

Field	Description
	10 Divide by 3 11 Divide by 4
23 DCU1_EN	DCU1 clock enable 0 Disable DCU1 clock 1 Enable DCU1 clock
22–20 DCU1_DIV	Divider to generate DCU1 clock 000 Divide by 1 001 Divide by 2 ... ... 111 Divide by 8
19 DCU0_EN	DCU0 clock enable 0 Disable DCU0 clock 1 Enable DCU0 clock
18–16 DCU0_DIV	Divider to generate DCU0 clock 000 Divide by 1 001 Divide by 2 ... ... 111 Divide by 8
15–13 NFC_PRE_DIV	NFC Pre-divider. The divider is used to provide divided clock to the fractional divider. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
12 QSPI1_EN	QSPI1 clock enable 0 Disable QSPI1 clock 1 Enable QSPI1 clock
11 QSPI1_DIV	QSPI1 clock divider 0 Divide by 1 1 Divide by 2
10 QSPI1_X2_DIV	QSPI1x2 Clock divider 0 Divide by 1 1 Divide by 2
9–8 QSPI1_X4_ DIV[1:0]	QSPI1x4 clock divider 00 Divide by 1 01 Divide by 2

Table continues on the next page...

CCM_CSCDR3 field descriptions (continued)

Field	Description
	10 Divide by 3 11 Divide by 4
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 QSPI0_EN	QSPI0 clock enable 0 Disable QSPI0 clock 1 Enable QSPI0 clock
3 QSPI0_DIV	QSPI0 clock divider 0 Divide by 1 1 Divide by 2
2 QSPI0_X2_DIV	QSPI0x2 clock divider 0 Divide by 1 1 Divide by 2
1–0 QSPI0_X4_DIV	QSPI0x4 clock divider 00 Divide by 1 01 Divide by 2 10 Divide by 3 11 Divide by 4

10.2.9 CCM Serial Clock Multiplexer Register 2 (CCM_CSCMR2)

The figure below represents the CCM Serial Clock Multiplexer Register 2 (CSCMR2). The CSCMR2 register contains bits to control the multiplexers that generate the module's clocks. The table below provides its field descriptions.

NOTE

Any change on the clock selection of the module must be done when the clock of the module is gated.

Address: 4006_B000h base + 20h offset = 4006_B020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												SWO_CLK_SEL	TRACE_CLK_SEL	FTM3_FIX_CLK_SEL	FTM2_FIX_CLK_SEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FTM1_FIX_CLK_SEL	FTM0_FIX_CLK_SEL	FTM3_EXT_CLK_SEL		FTM2_EXT_CLK_SEL		FTM1_EXT_CLK_SEL		FTM0_EXT_CLK_SEL		RMII_CLK_SEL		0	ENET_TS_CLK_SEL		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CSCMR2 field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 SWO_CLK_SEL	SWO clock select 0 24 MHz IRC clock 1 IPS Bus clock
18 TRACE_CLK_SEL	Trace clock select 0 Platform Bus clock 1 PLL3 Main clock
17 FTM3_FIX_CLK_SEL	FTM3 fixed clock select 0 32 KHz FXOSC clock 1 128 kHz IRC clock
16 FTM2_FIX_CLK_SEL	FTM2 fixed clock select 0 32 KHz FXOSC clock 1 128 kHz IRC clock
15 FTM1_FIX_CLK_SEL	FTM1 fixed clock select 0 32 KHz FXOSC clock 1 128 kHz IRC clock
14 FTM0_FIX_CLK_SEL	FTM0 fixed clock select 0 32 KHz FXOSC clock 1 128 kHz IRC clock
13–12 FTM3_EXT_CLK_SEL	FTM3 external clock select 00 128 kHz IRC clock 01 32 KHz FXOSC clock 10 12 MHz - 24 MHz FXOSC divided by 2 11 Audio External clock
11–10 FTM2_EXT_CLK_SEL	FTM2 external clock select 00 128 kHz IRC clock 01 32 KHz FXOSC clock 10 12 MHz - 24 MHz FXOSC divided by 2 11 Audio external clock

Table continues on the next page...

CCM_CSCMR2 field descriptions (continued)

Field	Description
9–8 FTM1_EXT_ CLK_SEL	FTM1 external clock select 00 128 kHz IRC clock 01 32 KHz FXOSC clock 10 12 MHz - 24 MHz FXOSC divided by 2 11 Audio external clock
7–6 FTM0_EXT_ CLK_SEL	FTM0 External clock select 00 128 kHz IRC clock 01 32 KHz FXOSC clock 10 12 MHz - 24 MHz FXOSC divided by 2 11 Audio external clock
5–4 RMII_CLK_SEL	Selector for ENET RMII clock multiplexer 00 ENET RMII clock 01 Audio external clock 10 PLL5 main clock 11 Divided by 2 of PLL5 main clock
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 ENET_TS_CLK_ SEL	Selector for ENET time sampling clock multiplexer 000 ENET RMII clock 001 24 MHz FXOSC clock 010 Audio external clock 011 USB clock (60 MHz) 100 ENET time stamping clock 101 PLL5 main clock divided by 2 110 PLL5 main clock

10.2.10 CCM Testing Observability Register (CCM_CTOR)

The figure below represents the CCM Testing Observability Register (CTOR). CCM includes three muxes to mux between different critical signals for testing observability. The output of the three muxes is generated on the three output signals obs_output_0, obs_output_1, and obs_output_2. Those three output signals can be generated on the IC pads by configuring the IOMUXC. The CTOR register contains bits to control the data generated for observability on the three output signals above. The table below provides its field descriptions.

Address: 4006_B000h base + 28h offset = 4006_B028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	OBS_EN	OBS_OUTPUT_2_SEL					OBS_OUTPUT_1_SEL					OBS_OUTPUT_0_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CTOR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 OBS_EN	observability enable bit. this bit enables the output of the three observability muxes. 0 Observability mux disabled. 1 Observability mux enabled.
14–10 OBS_OUTPUT_2_SEL	Selection of the signal to be generated on obs_output_2 (output of CCM) for observability on the pads. 00000 Core clock configuration in low power mode 00001 Reset clock enable 00010 PLL3 PFD3 disable 00011 PLL2 PFD3 disable 00100 PLL3 PFD4 disable 00101 PLL1 PFD3 disable 00110 PLL6 lock ready flag 00111 PLL3 lock ready flag

Table continues on the next page...

CCM_CTOR field descriptions (continued)

Field	Description	
	01000	Tied to Zero
	01001	Well Bias control
	01010	Tied to Zero
	01011	Low power mode FSM state [0]
	01100	Sampled Low power mode input [0]
	01101	low power handshake FSM state[0]
	01110	Low power mode request
	01111	Platform powerup request
	10000	FXOSC power down
	10001	Core Fuse
	10010	PLL enable
	10011	IPS wait request
	10100	Anatop Reg bypass
	10101	Core DSM request [0]
	10110	Tied to Zero
	10111	CAMP1 disable
	11000	Core DSM request[1]
	11001	Core clock stop
	11010	GPC clock synchronizer
	11011 - 11111	Tied to Zero
9–5 OBS_OUTPUT_1_SEL	Selection of the signal to be generated on obs_output_1 (output of CCM) for observability on the pads.	
	00000	Powerdown intiation indicator
	00001	Reset input
	00010	PLL3 PFD2 disable
	00011	PLL2 PFD2 disable
	00100	PLL2 PFD4 disable
	00101	PLL1 PFD2 disable
	00110	PLL5 Lock ready flag
	00111	PLL2 Lock ready flag
	01000	Tied to Zero
	01001	IPS debug enable
	01010	Tied to Zero
	01011	Low power mode FSM state [1]
	01100	Sampled Low power mode input [1]
	01101	low power handshake FSM state[1]
	01110	GPC Low power mode input
	01111	DAP power-up Ack
	10000	FIRX enable
	10001	Core Fuse [0]
	10010	PLL Ref enable
	10011	Stop mode indicator
	10100	System reset
	10101	Core Cortex-M4 clock gating enable
	10110	CAN 1 clock select
	10111	CAMP2 lock ready flag
	11000	Core DSM request [2]

Table continues on the next page...

CCM_CTOR field descriptions (continued)

Field	Description
	11001 clock gating mode 11010 IPS clock synchronizer 11011 - 11111 Tied to Zero
4–0 OBS_OUTPUT_ 0_SEL	Selection of the signal to be generated on obs_output_0 (output of CCM) for observability on the pads. 00000 Low power mode exit 00001 Clock ignition indicator 00010 PLL3 PFD1 disable 00011 PLL2 PFD1 disable 00100 PLL1 PFD4 disable 00101 PLL1 PFD1 disable 00110 PLL4 Lock ready flag 00111 PLL1 Lock ready flag 01000 Isolation enable 01001 IPS debug enable 01010 Memory repair mode 01011 Low power mode FSM state [2] 01100 Fuse latch/read indicator 01101 low power handshake FSM state[2] 01110 Deep sleep mode wakeup signal 01111 Debug power-up request 10000 FXOSC enable 10001 FUSE bits for core 10010 Wait mode indicator 10011 Stop mode indicator 10100 Camp disable 10101 Core Cortex-A5 clock gating enable 10110 CAN0 Clock Sel 10111 CAMP1 Lock ready Flag 11000 CORE DSM request [3] 11001 CORE Clock enable 11010 AHB clock Syncronizer 11011 - 11111 Tied to Zero

10.2.11 CCM Low Power Control Register (CCM_CLPCR)

The figure below represents the CCM Low Power Control Register (CLPCR). The CLPCR register contains bits to control the low power modes operation.The table below provides its field descriptions.

NOTE

Setting all the bits CCM_CLPCR[M_CORE0_WFI], CCM_CLPCR[M_CORE1_WFI], CCM_CLPCR[M_SCU_IDLE], and

CCM_CLPCR[M_L2CC_IDLE] will result in stop mode or low power stop mode depending upon GPC.

Address: 4006_B000h base + 2Ch offset = 4006_B02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						M_L2CC_IDLE	M_SCU_IDLE	M_CORE1_WFI	M_CORE0_WFI	0		Reserved	0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			FXOSC_BYPSS	FXOSC_PWRDWN	0	0	ANADIG_STOP_MODE	DIS_REF_OSC	SBYOS	ARM_CLK_LPM	Reserved		0		
W						FXOSC_BYPSEN										
Reset	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0

CCM_CLPCR field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 M_L2CC_IDLE	Mask L2CC IDLE for entering low power mode.

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
	0 L2CC IDLE is not masked 1 L2CC IDLE is masked
24 M_SCU_IDLE	Mask SCU IDLE for entering low power mode. 0 SCU IDLE is not masked 1 SCU IDLE is masked
23 M_CORE1_WFI	Mask WFI of core1 for entering low power mode 1 WFI of core1 is masked 0 WFI of core1 is not masked
22 M_CORE0_WFI	Mask WFI of core0 for entering low power mode 0 WFI of core0 is not masked 1 WFI of core0 is masked
21–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 Reserved	This field is reserved. Umimplemented
18–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 FXOSC_BYPSS	Bypass status of 24 MHz oscillator 0 24 Mhz oscialltor not bypassed 1 24 Mhz oscillator bypassed by external oscillator
11 FXOSC_ PWRDWN	In run mode, software can manually control powering down of on-chip oscillator. If software manually powered down the on-chip oscillator, then standby oscillator functionality for on-chip oscillator will be bypassed. The manual closing of on-chip oscillator should be performed only in case the reference oscilator is not the source of all the clocks generation. 0 On-chip oscillator will not be powered down 1 On-chip oscillator will be powered down
10 FXOSC_ BYPSEN	24 Mhz Oscillator bypass enable signal. 0 24 Mhz oscillator is not bypassed 1 24 Mhz oscillator is bypassed by external oscillator
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 ANADIG_STOP_ MODE	Masks the ANADIG stop mode indicator. See ANADIG_ANA_MISC0[STOP_MODE_CONFIG]. 0 Stop Mode indication is masked to Anatop 1 Stop Mode indication is sent to Anatop
7 DIS_REF_OSC	In run mode, software can manually control closing of external reference oscillator clock, i.e. generating '1' on ref_en_b signal. If software closed manually the external reference clock, then sbyos functionality will be bypassed. The manual closing of external reference oscilator should be performed only in case the reference oscilator is not the source of any clock generation.

Table continues on the next page...

CCM_CLPCR field descriptions (continued)

Field	Description
	0 External high frequency oscillator will be enabled, i.e. ref_en_b = '0'.(default) 1 External high frequency oscillator will be disabled, i.e. ref_en_b = '1'
6 SBYOS	Standby clock oscillator bit. This bit defines whether the 24 MHz FXOSC will be powered down in stop mode. This bit is discarded if CCM_CLPCR[FXOSC_PWRDWN] is asserted. 0 On-chip oscillator will not be powered down, after next entrance to stop mode. 1 On-chip oscillator will be powered down, after next entrance to stop mode.
5 ARM_CLK_LPM	Define if ARM clocks (Cortex-A5 and Cortex-M4) will be disabled on wait mode. This is useful for debug mode, when the user still wants to simulate entering wait mode and still keep ARM clock functioning. **NOTE:** Software should not enable ARM power gating in wait mode if this bit is cleared. 0 ARM clock enabled on wait mode. 1 ARM clock disabled on wait mode.
4–3 Reserved	This field is reserved. Reserved
2–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

10.2.12 CCM Interrupt Status Register (CCM_CISR)

The figure below represents the CCM Interrupt Status Register (CISR). This is a write one to clear register. Once a interrupt is generated, software should write one to clear it. The table below provides its field descriptions.

Address: 4006_B000h base + 30h offset = 4006_B030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									FXOSC_RDY	LRF_PLL6	LRF_PLL5	LRF_PLL4	LRF_PLL3	LRF_PLL2	LRF_PLL1
W										w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CISR field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FXOSC_RDY	Status of on baord oscilator ready, i.e. CCM_CCR[OSCNT] has finished counting. Interrupt will be generated when status is set. 0 On bard oscillatory is not ready 1 On bard oscillatory is ready
5 LRF_PLL6	Lock ready flag status of the PLL6 This is set only when the PLL6 is enabled and not when it is in bypass.

Table continues on the next page...

CCM_CISR field descriptions (continued)

Field	Description
	0 Lock ready flag of PLL6 generates no interrupt 1 Lock ready flag of PLL6 generates interrupt
4 LRF_PLL5	Lock ready flag status of the PLL5 This is set only when the PLL5 is enabled and not when it is in bypass. 0 Lock ready flag of PLL5 generates no interrupt 1 Lock ready flag of PLL5 generates interrupt
3 LRF_PLL4	Lock ready flag status of the PLL4 This is set only when the PLL5 is enabled and not when it is in bypass. 0 Lock ready flag of PLL4 generates no interrupt 1 Lock ready flag of PLL4 generates interrupt
2 LRF_PLL3	Lock ready flag status of the PLL3 This is set only when the PLL3 is enabled and not when it is in bypass. 0 Lock ready flag of PLL3 generates no interrupt 1 Lock ready flag of PLL3 generates interrupt
1 LRF_PLL2	Lock ready flag status of the PLL2 This is set only when the PLL2 is enabled and not when it is in bypass. 0 Lock ready flag of PLL2 generates no interrupt 1 Lock ready flag of PLL2 generates interrupt
0 LRF_PLL1	Lock ready flag status of the PLL1 This is set only when the PLL1 is enabled and not when it is in bypass. 0 Lock ready flag of PLL1 generates no interrupt 1 Lock ready flag of PLL1 generates interrupt

10.2.13 CCM Interrupt Mask Register (CCM_CIMR)

The figure below represents the CCM Interrupt Mask Register (CIMR). The table below provides its field descriptions.

Address: 4006_B000h base + 34h offset = 4006_B034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1									M_FXOSC_READY	M_LRF_PLL6	M_LRF_PLL5	M_LRF_PLL4	M_LRF_PLL3	M_LRF_PLL2	M_LRF_PLL1
W											w1c	w1c				
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CIMR field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
6 M_FXOSC_ READY	Mask interrupt generation due to on board oscillator ready 0 Interrupt is not masked 1 Interrupt is masked
5 M_LRF_PLL6	Mask interrupt generation due to lock ready flag of PLL6

Table continues on the next page...

CCM_CIMR field descriptions (continued)

Field	Description
	0 Interrupt is not masked 1 Interrupt is masked
4 M_LRF_PLL5	Mask interrupt generation due to lock ready flag of PLL5 0 Interrupt is not masked 1 Interrupt in masked
3 M_LRF_PLL4	Mask interrupt generation due to lock ready flag of PLL4 0 Interrupt is not masked 1 Interrupt is masked
2 M_LRF_PLL3	Mask interrupt generation due to lock ready flag of PLL3 0 Interrupt is not masked 1 Interrupt is masked
1 M_LRF_PLL2	Mask interrupt generation due to lock ready flag of PLL2 0 Interrupt is not masked 1 Interrupt is masked
0 M_LRF_PLL1	Mask interrupt generation due to lock ready flag of PLL1 0 Interrupt is not masked 1 Interrupt is masked

10.2.14 CCM Clock Output Source Register (CCM_CCOSR)

The figure below represents the CCM Clock Output Source Register (CCOSR). The CCOSR register contains bits to control the clocks that will be generated on the output CKO1 and CKO2.The table below provides its field descriptions.

Address: 4006_B000h base + 38h offset = 4006_B038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					CKO2_EN	CKO2_DIV				CKO2_SEL					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					CKO1_EN	CKO1_DIV				CKO1_SEL					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

CCM_CCOSR field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 CKO2_EN	Enable of CKO2 clock 0 CKO2 disabled. 1 CKO2 enabled.
25–22 CKO2_DIV	Setting the divider of CKO2 0000 Divide by 1(Default) 0001 Divide by 2 0010 Divide by 3 0011 Divide by 4 0100 Divide by 5 0101 Divide by 6 0110 Divide by 7 0111 Divide by 8 1000 Divide by 9 1001 Divide by 10 1010 Divide by 11 1011 Divide by 12 1100 Divide by 13 1101 Divide by 14 1110 Divide by 15 1111 Divide by 16
21–16 CKO2_SEL	Selection of the clock to be generated on CKO2 000000 Memory repair clock 000001 Video ADC divided by 2 clock 000010 Video ADC clock 000011 USB clock 000100 Trace clock 000101 SWO clock 000110 SPDIF recieve clock 000111 SPDIF clock 001000 SXOSC clock (synchronised at IPS Bus clock) 001001 SXOSC clock 001010 sirc_ipg_sync_clk 001011 SIRC clock (synchronised at IPS Bus clock) 001100 SAI3 clock 001101 SAI2 clock 001110 SAI1 clock 001111 SAI0 clock 010000 QSPI1 Serial x4 clock 010001 QSPI1 Serial x2 clock 010010 QSPI1 Serial clock 010011 Inverted QSPI1Serial clock 010100 QSPI0 Serial x4 clock

Table continues on the next page...

CCM_CCOSR field descriptions (continued)

Field	Description
	010101 QSPI0 Serial x2 clock 010110 PLL6 Main clock 010111 PLL6 Divided by 2 clock 011000 PLL5 Main clock 011001 PLL5 Divided clock 011010 PLL4 Main clock 011011 PLL4 Divided clock 011100 PLL3 PFD4 clock 011101 PLL3 PFD3 clock 011110 PLL3 PFD2 clock 011111 PLL3 PFD1 clock 100000 PLL3 Main clock 100001 PLL3 divided clock 100010 PLL2 PFD4 clock 100011 PLL2 PFD3 clock 100100 PLL2 PFD2 clock 100101 PLL2 PFD1 clock 100110 PLL2 Main clock 100111 PLL1 PFD4 clock 101000 PLL1 PFD3 clock 101001 PLL1 PFD2 clock 101010 PLL1 PFD1 clock 101011 PLL1 Main clock 101100 - 111111 Tied to Zero
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 CKO1_EN	Enable of CKO1 clock 0 CKO1 disabled. 1 CKO1 enabled.
9–6 CKO1_DIV	Setting the divider of CKO1 0000 Divide by 1(default) 0001 Divide by 2 0010 Divide by 3 0011 Divide by 4 0100 Divide by 5 0101 Divide by 6 0110 Divide by 7 0111 Divide by 8 1000 Divide by 9 1001 Divide by 10 1010 Divide by 11 1011 Divide by 12 1100 Divide by 13 1101 Divide by 14

Table continues on the next page...

CCM_CCOSR field descriptions (continued)

Field	Description
	1110 Divide by 15 1111 Divide by 16
5–0 CKO1_SEL	Selection of the clock to be generated on CKO1 000000 QSPI 0 Serial clk 000001 QSPI 0 Serial clk -inverted 000010 PLL6 Main clock 000011 PLL6 Main Clock divided by 2 000100 PLL5 Main Clock 000101 PLL5 divided Clock 000110 PLL4 Main clock 000111 PLL4 Divided Clock 001000 PLL3 PFD 4 clock 001001 PLL3 PFD 3 clock 001010 PLL3 PFD 2 clock 001011 PLL3 PFD 1 clock 001100 PLL3 Main clock 001101 PLL3 Div clock 001110 PLL2 PFD 4 clock 001111 PLL2 PFD 3 clock 010000 PLL2 PFD 2 clock 010001 PLL2 PFD 1 clock 010010 PLL2 Main clock 010011 PLL1 PFD 4 clock 010100 PLL1 PFD 3 clock 010101 PLL1 PFD 2 clock 010110 PLL1 PFD 1 clock 010111 PLL1 Main clock 011000 nfc_clk_root 011001 Reserved 011010 Test Clock 0 011011 IPS Bus clock 011100 ENET Time Sampling clock 011101 Reserved 011110 GPC clock 011111 FTM3 Fix clock 100000 FTM3 External clock 100001 FTM2 Fix clock 100010 FTM2 External clock 100011 FTM1 Fix clock 100100 FTM1 External clock 100101 FTM0 Fix clock 100110 FTM0 External clock 100111 FXOSC Divided by 2 clock 101000 FXOSC clock 101001 FIRC clock 101010 ESDHC1 clock

Table continues on the next page...

CCM_CCOSR field descriptions (continued)

Field	Description
	101011 ESDHC0 clock 101100 ESAI clock 101101 ENET External clock (scan muxed) 101110 ENET TS input clock 101111 ENET RMII clock 110000 ENET External clock 110001 Scan clock 110010 TCLK 110011 DRAMC clock 110100 DCU1 pixel clock 110101 DCU0 pixel clock 110110 DAP clock 110111 Cortex-M4 core clock 111000 32 KHz Clock 111001 24Mhz clock 111010 Clock Igniton clock 111011 CAN1 clock 111100 CAN0 clock 111101 Cortex-A5 core clock 111110 Platform Bus clock 111111 Audio external clock

10.2.15 CCM General Purpose Register (CCM_CGPR)

The figure below represents the CCM General Purpose Register (CGPR). The table below provides its field descriptions.

Address: 4006_B000h base + 3Ch offset = 4006_B03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											0	0		QSPIn_ACCZ	FS_ENABLE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CGPR field descriptions

Field	Description
31–5 Reserved	This field is reserved. Reserved for future use.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 QSPIn_ACCZ	Used for passing bus clock as source to QSPI. It overrides CCM_CSCMR1[QSPIn_CLK_SEL]. 0 Clock source to QSPIn depends on CCM_CSCMR1[QSPIn_CLK_SEL] 1 QSPIn clock source is platform bus clock
0 FS_ENABLE	MLB 1024xFs enable 0 1024xFs MLB is disabled 1 1024xFs is enabled

10.2.16 CCM Clock Gating Register (CCM_CCGR*n*)

The figure below represents the CCM Clock Gating Register (CCM_CCGR). These registers define the clock gating for power reduction of each module as shown in Table 10-20 (CG(i) bits). Each CG(i) field consists of two bits. There are 12 CGR registers. The number of registers required is according to the number of peripherals in the system.

NOTE

For modules on multiple AIPS slots, only the first address is used for clock gating.

Table 10-20. CCGR mapping table

Register	AIPS0/1 Slot #	Module Controlled
CCM_CCGR0[CG0]	AIPS0-Slot32	FlexCAN0
CCM_CCGR0[CG1]	AIPS0-Slot33	
CCM_CCGR0[CG2]	AIPS0-Slot34	
CCM_CCGR0[CG3]	AIPS0-Slot35	
CCM_CCGR0[CG4]	AIPS0-Slot36	DMA channel Mux0
CCM_CCGR0[CG5]	AIPS0-Slot37	DMA channel Mux1
CCM_CCGR0[CG6]	AIPS0-Slot38	
CCM_CCGR0[CG7]	AIPS0-Slot39	UART0
CCM_CCGR0[CG8]	AIPS0-Slot40	UART1
CCM_CCGR0[CG9]	AIPS0-Slot41	UART2
CCM_CCGR0[CG10]	AIPS0-Slot42	UART3
CCM_CCGR0[CG11]	AIPS0-Slot43	
CCM_CCGR0[CG12]	AIPS0-Slot44	SPI 0

Table continues on the next page...

Table 10-20. CCGR mapping table (continued)

Register	AIPS0/1 Slot #	Module Controlled
CCM_CCGR0[CG13]	AIPS0-Slot45	SPI 1
CCM_CCGR0[CG14]	AIPS0-Slot46	
CCM_CCGR0[CG15]	AIPS0-Slot47	SAI0
CCM_CCGR1[CG16]	AIPS0-Slot48	SAI1
CCM_CCGR1[CG17]	AIPS0-Slot49	SAI2
CCM_CCGR1[CG18]	AIPS0-Slot50	SAI3
CCM_CCGR1[CG19]	AIPS0-Slot51	CRC
CCM_CCGR1[CG20]	AIPS0-Slot52	USBC0
CCM_CCGR1[CG21]	AIPS0-Slot53	
CCM_CCGR1[CG22]	AIPS0-Slot54	Programmable delay block (PDB)
CCM_CCGR1[CG23]	AIPS0-Slot55	Periodic interrupt timers (PIT)
CCM_CCGR1[CG24]	AIPS0-Slot56	FlexTimer (FTM 0)
CCM_CCGR1[CG25]	AIPS0-Slot57	FlexTimer (FTM 1)
CCM_CCGR1[CG26]	AIPS0-Slot58	
CCM_CCGR1[CG27]	AIPS0-Slot59	ADC0
CCM_CCGR1[CG28]	AIPS0-Slot60	
CCM_CCGR1[CG29]	AIPS0-Slot61	TCON0
CCM_CCGR1[CG30]	AIPS0-Slot62	WDOG-A5
CCM_CCGR1[CG31]	AIPS0-Slot63	WDOG-M4
CCM_CCGR2[CG32]	AIPS0-Slot64	Low-power timer (LPTMR)
CCM_CCGR2[CG33]	AIPS0-Slot65	
CCM_CCGR2[CG34]	AIPS0-Slot66	RLE
CCM_CCGR2[CG35]	AIPS0-Slot67	Reserved
CCM_CCGR2[CG36]	AIPS0-Slot68	QuadSPI0
CCM_CCGR2[CG37]	AIPS0-Slot69	
CCM_CCGR2[CG38]	AIPS0-Slot70	
CCM_CCGR2[CG39]	AIPS0-Slot71	
CCM_CCGR2[CG40]	AIPS0-Slot72	IOMUX Controller
CCM_CCGR2[CG41]	AIPS0-Slot73	Port A multiplexing control
CCM_CCGR2[CG42]	AIPS0-Slot74	Port B multiplexing control
CCM_CCGR2[CG43]	AIPS0-Slot75	Port C multiplexing control
CCM_CCGR2[CG44]	AIPS0-Slot76	Port D multiplexing control
CCM_CCGR2[CG45]	AIPS0-Slot77	Port E multiplexing control
CCM_CCGR2[CG46]	AIPS0-Slot78	
CCM_CCGR2[CG47]	AIPS0-Slot79	
CCM_CCGR3[CG48]	AIPS0-Slot80	ANADIG
CCM_CCGR3[CG49]	AIPS0-Slot81	
CCM_CCGR3[CG50]	AIPS0-Slot82	Slow Clock Source Controller Module (SCSCM)
CCM_CCGR3[CG51]	AIPS0-Slot83	

Table continues on the next page...

Table 10-20. CCGR mapping table (continued)

Register	AIPS0/1 Slot #	Module Controlled
CCM_CCGR3[CG52]	AIPS0-Slot84	
CCM_CCGR3[CG53]	AIPS0-Slot85	
CCM_CCGR3[CG54]	AIPS0-Slot86	
CCM_CCGR3[CG55]	AIPS0-Slot87	
CCM_CCGR3[CG56]	AIPS0-Slot88	DCU0
CCM_CCGR3[CG57]	AIPS0-Slot89	
CCM_CCGR3[CG58]	AIPS0-Slot90	
CCM_CCGR3[CG59]	AIPS0-Slot91	
CCM_CCGR3[CG60]	AIPS0-Slot92	
CCM_CCGR3[CG61]	AIPS0-Slot93	
CCM_CCGR3[CG62]	AIPS0-Slot94	
CCM_CCGR3[CG63]	AIPS0-Slot95	
CCM_CCGR4[CG64]	AIPS0-Slot96	ASRC
CCM_CCGR4[CG65]	AIPS0-Slot97	SPDIF
CCM_CCGR4[CG66]	AIPS0-Slot98	ESAI
CCM_CCGR4[CG67]	AIPS0-Slot99	
CCM_CCGR4[CG68]	AIPS0-Slot100	
CCM_CCGR4[CG69]	AIPS0-Slot101	External watchdog (EWM)
CCM_CCGR4[CG70]	AIPS0-Slot102	I2C 0
CCM_CCGR4[CG71]	AIPS0-Slot103	I2C 1
CCM_CCGR4[CG72]	AIPS0-Slot104	
CCM_CCGR4[CG73]	AIPS0-Slot105	
CCM_CCGR4[CG74]	AIPS0-Slot106	Wake up Unit (WKUP)
CCM_CCGR4[CG75]	AIPS0-Slot107	Clock Control Module (CCM)
CCM_CCGR4[CG76]	AIPS0-Slot108	Global Power Controller (GPC)
CCM_CCGR4[CG77]	AIPS0-Slot109	Voltage Regulator-Digital (VREG_DIG)
CCM_CCGR4[CG78]	AIPS0-Slot110	Reserved
CCM_CCGR4[CG79]	AIPS0-Slot111	Clock Monitor Unit (CMU)

Table continues on the next page...

Table 10-20. CCGR mapping table (continued)

Register	AIPS0/1 Slot #	Module Controlled
CCM_CCGR5[CG80]	Reserved / Not used	
CCM_CCGR5[CG81]		
CCM_CCGR5[CG82]		
CCM_CCGR5[CG83]		
CCM_CCGR5[CG84]		
CCM_CCGR5[CG85]		
CCM_CCGR5[CG86]		
CCM_CCGR5[CG87]		
CCM_CCGR5[CG88]		
CCM_CCGR5[CG89]		
CCM_CCGR5[CG90]		
CCM_CCGR5[CG91]		
CCM_CCGR5[CG92]		
CCM_CCGR5[CG93]		
CCM_CCGR5[CG94]		
CCM_CCGR5[CG95]		
CCM_CCGR6[CG96]	AIPS1-Slot32	
CCM_CCGR6[CG97]	AIPS1-Slot33	DMA Channel Mux2
CCM_CCGR6[CG98]	AIPS1-Slot34	DMA Channel Mux3
CCM_CCGR6[CG99]	AIPS1-Slot35	
CCM_CCGR6[CG100]	AIPS1-Slot36	
CCM_CCGR6[CG101]	AIPS1-Slot37	OTP CTRL
CCM_CCGR6[CG102]	AIPS1-Slot38	
CCM_CCGR6[CG103]	AIPS1-Slot39	Reserved
CCM_CCGR6[CG104]	AIPS1-Slot40	Reserved
CCM_CCGR6[CG105]	AIPS1-Slot41	UART4
CCM_CCGR6[CG106]	AIPS1-Slot42	UART5
CCM_CCGR6[CG107]	AIPS1-Slot43	
CCM_CCGR6[CG108]	AIPS1-Slot44	SPI 2
CCM_CCGR6[CG109]	AIPS1-Slot45	SPI 3
CCM_CCGR6[CG110]	AIPS1-Slot46	DDRMC[1]
CCM_CCGR6[CG111]	AIPS1-Slot47	
CCM_CCGR7[CG112]	AIPS1-Slot48	
CCM_CCGR7[CG113]	AIPS1-Slot49	SDHC0
CCM_CCGR7[CG114]	AIPS1-Slot50	SDHC1
CCM_CCGR7[CG115]	AIPS1-Slot51	
CCM_CCGR7[CG116]	AIPS1-Slot52	USBC1
CCM_CCGR7[CG117]	AIPS1-Slot53	
CCM_CCGR7[CG118]	AIPS1-Slot54	

Table continues on the next page...

Table 10-20. CCGR mapping table (continued)

Register	AIPS0/1 Slot #	Module Controlled
CCM_CCGR7[CG119]	AIPS1-Slot55	
CCM_CCGR7[CG120]	AIPS1-Slot56	FlexTimer (FTM) 2
CCM_CCGR7[CG121]	AIPS1-Slot57	FlexTimer (FTM) 3
CCM_CCGR7[CG122]	AIPS1-Slot58	
CCM_CCGR7[CG123]	AIPS1-Slot59	ADC 1
CCM_CCGR7[CG124]	AIPS1-Slot60	
CCM_CCGR7[CG125]	AIPS1-Slot61	TCON1
CCM_CCGR7[CG126]	AIPS1-Slot62	Segment LCD
CCM_CCGR7[CG127]	AIPS1-Slot63	
CCM_CCGR8[CG128]	AIPS1-Slot64	
CCM_CCGR8[CG129]	AIPS1-Slot65	
CCM_CCGR8[CG130]	AIPS1-Slot66	
CCM_CCGR8[CG131]	AIPS1-Slot67	
CCM_CCGR8[CG132]	AIPS1-Slot68	QuadSPI1
CCM_CCGR8[CG133]	AIPS1-Slot69	
CCM_CCGR8[CG134]	AIPS1-Slot70	
CCM_CCGR8[CG135]	AIPS1-Slot71	Video ADC
CCM_CCGR8[CG136]	AIPS1-Slot72	Video Decoder
CCM_CCGR8[CG137]	AIPS1-Slot73	VIU3
CCM_CCGR8[CG138]	AIPS1-Slot74	
CCM_CCGR8[CG139]	AIPS1-Slot75	
CCM_CCGR8[CG140]	AIPS1-Slot76	12-bit digital-to-analog converter (DAC) 0
CCM_CCGR8[CG141]	AIPS1-Slot77	12-bit digital-to-analog converter (DAC) 1
CCM_CCGR8[CG142]	AIPS1-Slot78	
CCM_CCGR9[CG144]	AIPS1-Slot80	Ethernet MAC0 and IEEE 1588 timers
CCM_CCGR9[CG145]	AIPS1-Slot81	Ethernet MAC1 and IEEE 1588 timers
CCM_CCGR9[CG146]	AIPS1-Slot82	
CCM_CCGR9[CG147]	AIPS1-Slot83	
CCM_CCGR9[CG148]	AIPS1-Slot84	FlexCAN1
CCM_CCGR9[CG149]	AIPS1-Slot85	
CCM_CCGR9[CG150]	AIPS1-Slot86	
CCM_CCGR9[CG151]	AIPS1-Slot87	
CCM_CCGR9[CG152]	AIPS1-Slot88	DCU1
CCM_CCGR9[CG153]	AIPS1-Slot89	
CCM_CCGR9[CG154]	AIPS1-Slot90	
CCM_CCGR9[CG155]	AIPS1-Slot91	
CCM_CCGR9[CG156]	AIPS1-Slot92	
CCM_CCGR9[CG157]	AIPS1-Slot93	

Table continues on the next page...

Table 10-20. CCGR mapping table (continued)

Register	AIPS0/1 Slot #	Module Controlled
CCM_CCGR9[CG158]	AIPS1-Slot94	
CCM_CCGR9[CG159]	AIPS1-Slot95	
CCM_CCGR10[CG160]	AIPS1-Slot96	Nand Flash Controller (NFC)
CCM_CCGR10[CG161]	AIPS1-Slot97	
CCM_CCGR10[CG162]	AIPS1-Slot98	
CCM_CCGR10[CG163]	AIPS1-Slot99	
CCM_CCGR10[CG164]	AIPS1-Slot100	
CCM_CCGR10[CG165]	AIPS1-Slot101	
CCM_CCGR10[CG166]	AIPS1-Slot102	I2C2
CCM_CCGR10[CG167]	AIPS1-Slot103	I2C3
CCM_CCGR10[CG168]	AIPS1-Slot104	Ethernet L2 Switch
CCM_CCGR10[CG169]	AIPS1-Slot105	
CCM_CCGR10[CG170]	AIPS1-Slot106	
CCM_CCGR10[CG171]	AIPS1-Slot107	
CCM_CCGR10[CG172]	AIPS1-Slot108	
CCM_CCGR10[CG173]	AIPS1-Slot109	
CCM_CCGR10[CG174]	AIPS1-Slot110	
CCM_CCGR10[CG175]	AIPS1-Slot111	
CCM_CCGR11[CG176]	AIPS1-Slot112	Reserved
CCM_CCGR11[CG177]	AIPS1-Slot113	
CCM_CCGR11[CG178]	AIPS1-Slot114	
CCM_CCGR11[CG179]	AIPS1-Slot115	
CCM_CCGR11[CG180]	AIPS1-Slot116	
CCM_CCGR11[CG181]	AIPS1-Slot117	
CCM_CCGR11[CG182]	AIPS1-Slot118	
CCM_CCGR11[CG183]	AIPS1-Slot119	
CCM_CCGR11[CG184]	AIPS1-Slot120	
CCM_CCGR11[CG185]	AIPS1-Slot121	
CCM_CCGR11[CG186]	AIPS1-Slot122	
CCM_CCGR11[CG187]	AIPS1-Slot123	
CCM_CCGR11[CG188]	AIPS1-Slot124	
CCM_CCGR11[CG189]	AIPS1-Slot125	
CCM_CCGR11[CG190]	AIPS1-Slot126	
CCM_CCGR11[CG191]	AIPS1-Slot127	

1. Register read or write for any register in the DDR controller requires the DDRMC clock to be configured from CCM_CCSR[DDRC_CLK_SEL] in addition to enabling the clock from the clock gating register: CCM_CCGR6[CG110]

Table 10-21. CG Bit Description

CGR value	Clock Activity Description
00	Clock is off during all modes. stop enter hardware handshake is disabled.
01	Clock is on in run mode, but off in wait and stop modes.
10	Clock is on during all modes, even stop mode.
11	Clock is on during all modes, except stop mode.

Address: 4006_B000h base + 40h offset + (4d × i), where i=0d to 11d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CG15		CG14		CG13		CG12		CG11		CG10		CG9		CG8	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CG7		CG6		CG5		CG4		CG3		CG2		CG1		CG0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CCGR*n* field descriptions

Field	Description
31–30 CG15	Clock gating enable.
29–28 CG14	Clock gating enable.
27–26 CG13	Clock gating enable.
25–24 CG12	Clock gating enable.
23–22 CG11	Clock gating enable.
21–20 CG10	Clock gating enable.
19–18 CG9	Clock gating enable.
17–16 CG8	Clock gating enable.
15–14 CG7	Clock gating enable.
13–12 CG6	Clock gating enable.
11–10 CG5	Clock gating enable.
9–8 CG4	Clock gating enable.
7–6 CG3	Clock gating enable.

Table continues on the next page...

CCM_CCGR*n* field descriptions (continued)

Field	Description
5–4 CG2	Clock gating enable.
3–2 CG1	Clock gating enable.
1–0 CG0	Clock gating enable.

1. Register read or write for any register in the DDR controller requires the DDRMC clock to be configured from CCM_CCSR[DDRC_CLK_SEL] in addition to enabling the clock from the clock gating register: CCM_CCGR6[CG110]

10.2.17 CCM Module Enable Override Register (CCM_CMEOR*n*)

The following figure represents the CCM Module Enable Override Register (CMEOR). The CMEOR register contains bits to override the clock enable signal from the module. These bits are applicable for only those modules whose clocks are enabled by clock enable signal.

MO(i) fields are used to override the clock enable signal from each module independently.

The register mapping (clock enabling conditions for each module) for the different MOR's is same as that for different CGR's, as shown in Table 10-20

NOTE

For modules on multiple AIPS slots, only the first address is used for module enabling.

Table 10-59. MO Bitfield Descriptions

MO Value	Description
0	Don’t override module enable signal
1	Override module enable signal

NOTE

For description of individual field, refer to Peripheral Bridge 0 (AIPS-Lite 0) Memory Map and Peripheral Bridge 1 (AIPS-Lite 1) Memory Map

Address: 4006_B000h base + 70h offset + (4d × i), where i=0d to 5d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	MO31	MO30	MO29	MO28	MO27	MO26	MO25	MO24	MO23	MO22	MO21	MO20	MO19	MO18	MO17	MO16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	MO15	MO14	MO13	MO12	MO11	MO10	MO9	MO8	MO7	MO6	MO5	MO4	MO3	MO2	MO1	MO0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CMEOR*n* field descriptions

Field	Description
31 MO31	Override module enable
30 MO30	Override module enable
29 MO29	Override module enable
28 MO28	Override module enable
27 MO27	Override module enable
26 MO26	Override module enable
25 MO25	Override module enable
24 MO24	Override module enable
23 MO23	Override module enable
22 MO22	Override module enable
21 MO21	Override module enable
20 MO20	Override module enable
19 MO19	Override module enable
18 MO18	Override module enable
17 MO17	Override module enable

Table continues on the next page...

CCM_CMEOR*n* field descriptions (continued)

Field	Description
16 MO16	Override module enable
15 MO15	Override module enable
14 MO14	Override module enable
13 MO13	Override module enable
12 MO12	Override module enable
11 MO11	Override module enable
10 MO10	Override module enable
9 MO9	Override module enable
8 MO8	Override module enable
7 MO7	Override module enable
6 MO6	Override module enable
5 MO5	Override module enable
4 MO4	Override module enable
3 MO3	Override module enable
2 MO2	Override module enable
1 MO1	Override module enable
0 MO0	Override module enable

10.2.18 CCM PLL PFD Disable Status Register (CCM_CPPDSR)

Address: 4006_B000h base + 88h offset = 4006_B088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				PLL3_PFD4	PLL3_PFD3	PLL3_PFD2	PLL3_PFD1	PLL2_PFD4	PLL2_PFD3	PLL2_PFD2	PLL2_PFD1	PLL1_PFD4	PLL1_PFD3	PLL1_PFD2	PLL1_PFD1
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

CCM_CPPDSR field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 PLL3_PFD4	Indicates the status of PLL3 PFD4 disable signal 0 PFD output is enabled 1 PFD output is disabled
10 PLL3_PFD3	Indicates the status of PLL3 PFD3 disable signal 0 PFD output is enabled 1 PFD output is disabled
9 PLL3_PFD2	Indicates the status of PLL3 PFD2 disable signal

Table continues on the next page...

CCM_CPPDSR field descriptions (continued)

Field	Description
	0 PFD output is enabled 1 PFD output is disabled
8 PLL3_PFD1	Indicates the status of PLL3 PFD1 disable signal 0 PFD output is enabled 1 PFD output is disabled
7 PLL2_PFD4	Indicates the status of PLL2 PFD4 disable signal 0 PFD output is enabled 1 PFD output is disabled
6 PLL2_PFD3	Indicates the status of PLL2 PFD3 disable signal 0 PFD output is enabled 1 PFD output is disabled
5 PLL2_PFD2	Indicates the status of PLL2 PFD2 disable signal 0 PFD output is enabled 1 PFD output is disabled
4 PLL2_PFD1	Indicates the status of PLL2 PFD1 disable signal 0 PFD output is enabled 1 PFD output is disabled
3 PLL1_PFD4	Indicates the status of PLL1 PFD4 disable signal 0 PFD output is enabled 1 PFD output is disabled
2 PLL1_PFD3	Indicates the status of PLL1 PFD3 disable signal 0 PFD output is enabled 1 PFD output is disabled
1 PLL1_PFD2	Indicates the status of PLL1 PFD2 disable signal 0 PFD output is enabledPFD output is enabled 1 PFD output is disabled
0 PLL1_PFD1	Indicates the status of PLL1 PFD1 disable signal 0 PFD output is enabled 1 PFD output is disabledPFD output is disabled

10.2.19 CCM CORE Wakeup Register (CCM_CCOWR)

The figure represents the CCM Core Wakeup Register (CCOWR).

Address: 4006_B000h base + 8Ch offset = 4006_B08Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															AUX_CORE_WKUP
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	WAKEUP_KEY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_CCOWR field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 AUX_CORE_ WKUP	Wakes up the secondary (non-master) core. 0 Secondary core clock is gated 1 Secondary core clock is enabled
15–0 WAKEUP_KEY	This should be written with key 5A5Ah in order to write to the AUX_CORE_WKUP.

10.2.20 CCM Platform Clock Gating Register (CCM_CCPGR*n*)

The following figure represents the CCM Platform Clock Gating Register (CCPGR). The clock gating Registers define the clock gating for power reduction of each clock (PCG(i) bits). There are four CCPGR registers. The number of registers required is according to the number of peripherals in the system.

Table 10-75. CCPGR mapping table

Registers	AIPS0/1 Slot #	Module Controlled
CCM_CCPGR0[PPCG0]	AIPS0-Slot0	
CCM_CCPGR0[PPCG1]	AIPS0-Slot1	MSCM (CPU Configuration, INTR, OCRAM)
CCM_CCPGR0[PPCG2]	AIPS0-Slot2	Cortex-A5-SCU+GIC
CCM_CCPGR0[PPCG3]	AIPS0-Slot3	Cortex-A5-INTD
CCM_CCPGR0[PPCG4]	AIPS0-Slot4	
CCM_CCPGR0[PPCG5]	AIPS0-Slot5	
CCM_CCPGR0[PPCG6]	AIPS0-Slot6	Cortex-A5-L2C (Cortex-A5 L2 Cache Controller)
CCM_CCPGR0[PPCG7]	AIPS0-Slot7	
CCM_CCPGR0[PPCG8]	AIPS0-Slot8	NIC0 (Network Interconnect)
CCM_CCPGR0[PPCG9]	AIPS0-Slot9	NIC1
CCM_CCPGR0[PPCG10]	AIPS0-Slot10	NIC2
CCM_CCPGR0[PPCG11]	AIPS0-Slot11	NIC3
CCM_CCPGR0[PPCG12]	AIPS0-Slot12	NIC4
CCM_CCPGR0[PPCG13]	AIPS0-Slot13	NIC5
CCM_CCPGR0[PPCG14]	AIPS0-Slot14	NIC6
CCM_CCPGR0[PPCG15]	AIPS0-Slot15	NIC7
CCM_CCPGR1[PPCG16]	AIPS0-Slot16	Reserved
CCM_CCPGR1[PPCG17]	AIPS0-Slot17	Reserved
CCM_CCPGR1[PPCG18]	AIPS0-Slot18	Reserved
CCM_CCPGR1[PPCG19]	AIPS0-Slot19	Reserved
CCM_CCPGR1[PPCG20]	AIPS0-Slot20	Reserved
CCM_CCPGR1[PPCG21]	AIPS0-Slot21	Reserved
CCM_CCPGR1[PPCG22]	AIPS0-Slot22	
CCM_CCPGR1[PPCG23]	AIPS0-Slot23	Reserved
CCM_CCPGR1[PPCG24]	AIPS0-Slot24	DMA0
CCM_CCPGR1[PPCG25]	AIPS0-Slot25	DMA0_TCD
CCM_CCPGR1[PPCG26]	AIPS0-Slot26	
CCM_CCPGR1[PPCG27]	AIPS0-Slot27	
CCM_CCPGR1[PPCG28]	AIPS0-Slot28	
CCM_CCPGR1[PPCG29]	AIPS0-Slot29	SEMA4

Table continues on the next page...

Table 10-75. CCPGR mapping table (continued)

Registers	AIPS0/1 Slot #	Module Controlled
CCM_CCPGR1[PPCG30]	AIPS0-Slot30	FlexBus
CCM_CCPGR1[PPCG31]	AIPS0-Slot31	
CCM_CCPGR2[PPCG32]	AIPS1-Slot0	
CCM_CCPGR2[PPCG33]	AIPS1-Slot1	
CCM_CCPGR2[PPCG34]	AIPS1-Slot2	
CCM_CCPGR2[PPCG35]	AIPS1-Slot3	
CCM_CCPGR2[PPCG36]	AIPS1-Slot4	
CCM_CCPGR2[PPCG37]	AIPS1-Slot5	
CCM_CCPGR2[PPCG38]	AIPS1-Slot6	
CCM_CCPGR2[PPCG39]	AIPS1-Slot7	Debug Access Port (DAP)-RomTable
CCM_CCPGR2[PPCG40]	AIPS1-Slot8	Cortex-A5-DBG (Cortex-A5 Debug)
CCM_CCPGR2[PPCG41]	AIPS1-Slot9	Cortex-A5-PMU (Cortex-A5 Performance Monitoring Unit)
CCM_CCPGR2[PPCG42]	AIPS1-Slot10	Cortex-A5-ETM
CCM_CCPGR2[PPCG43]	AIPS1-Slot11	
CCM_CCPGR2[PPCG44]	AIPS1-Slot12	Cortex-A5-RomTable
CCM_CCPGR2[PPCG45]	AIPS1-Slot13	
CCM_CCPGR2[PPCG46]	AIPS1-Slot14	Cortex-A5-CTI
CCM_CCPGR2[PPCG47]	AIPS1-Slot15	
CCM_CCPGR3[PPCG48]	AIPS1-Slot16	Cortex-A5-ITM
CCM_CCPGR3[PPCG49]	AIPS1-Slot17	Cortex-A5-ETB
CCM_CCPGR3[PPCG50]	AIPS1-Slot18	Cortex-A5-Funnel
CCM_CCPGR3[PPCG51]	AIPS1-Slot19	Pltf-TCTL
CCM_CCPGR3[PPCG52]	AIPS1-Slot20	Pltf-TPIU
CCM_CCPGR3[PPCG53]	AIPS1-Slot21	Pltf-Funnel
CCM_CCPGR3[PPCG54]	AIPS1-Slot22	Pltf-SWO
CCM_CCPGR3[PPCG55]	AIPS1-Slot23	
CCM_CCPGR3[PPCG56]	AIPS1-Slot24	DMA1
CCM_CCPGR3[PPCG57]	AIPS1-Slot25	DMA1_TCD
CCM_CCPGR3[PPCG58]	AIPS1-Slot26	
CCM_CCPGR3[PPCG59]	(AIPS1-Slot27)	
CCM_CCPGR3[PPCG60]	(AIPS1-Slot28)	
CCM_CCPGR3[PPCG61]	(AIPS1-Slot29)	
CCM_CCPGR3[PPCG62]	(AIPS1-Slot30)	
CCM_CCPGR3[PPCG63]	(AIPS1-Slot31)	

Table 10-76. PPCG Bit Description

PPCGR Value	Clock Activity Description
00	Clock is off during all modes.

Table continues on the next page...

Table 10-76. PPCG Bit Description (continued)

PPCGR Value	Clock Activity Description
01	Clock is on in run mode, but off in wait and stop modes.
10	Clock is on during all modes, even stop mode.
11	Clock is on during all modes except stop mode.

Address: 4006_B000h base + 90h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPCG15		PPCG14		PPCG13		PPCG12		PPCG11		PPCG10		PPCG9		PPCG8	
Reset	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPCG7		PPCG6		PPCG5		PPCG4		PPCG3		PPCG2		PPCG1		PPCG0	
Reset	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*	1*

* Notes:
- CCPGR1 register reset value is CFFF_FFFFh

CCM_CCPGR*n* field descriptions

Field	Description
31–30 PPCG15	Platform Peripheral gating definition bits
29–28 PPCG14	Platform Peripheral gating definition bits
27–26 PPCG13	Platform Peripheral gating definition bits
25–24 PPCG12	Platform Peripheral gating definition bits
23–22 PPCG11	Platform Peripheral gating definition bits
21–20 PPCG10	Platform Peripheral gating definition bits
19–18 PPCG9	Platform Peripheral gating definition bits
17–16 PPCG8	Platform Peripheral gating definition bits
15–14 PPCG7	Platform Peripheral gating definition bits
13–12 PPCG6	Platform Peripheral gating definition bits
11–10 PPCG5	Platform Peripheral gating definition bits
9–8 PPCG4	Platform Peripheral gating definition bits

Table continues on the next page...

CCM_CCPGR*n* field descriptions (continued)

Field	Description
7–6 PPCG3	Platform Peripheral gating definition bits
5–4 PPCG2	Platform Peripheral gating definition bits
3–2 PPCG1	Platform Peripheral gating definition bits
1–0 PPCG0	Platform Peripheral gating definition bits

10.3 Sync Signals

Sync signals are generated for edge alignment.

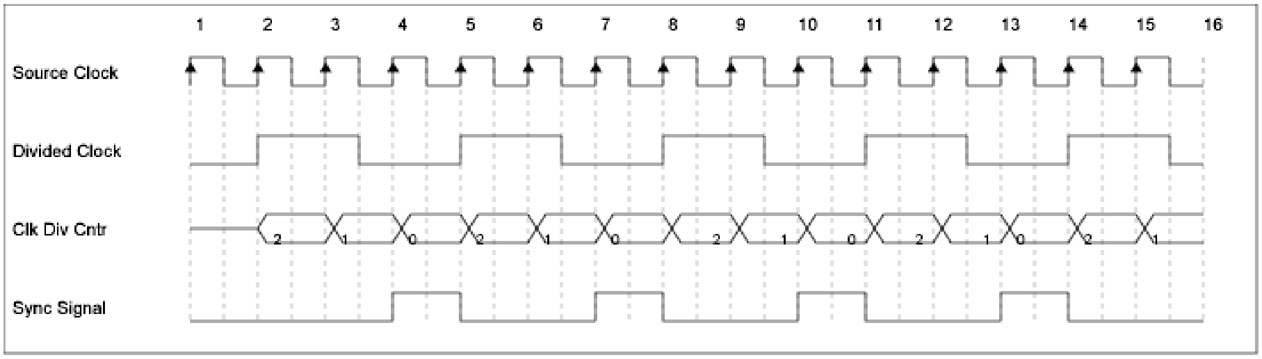


Figure 10-44. Sync Signals

10.4 Accessory clocks

For all the Accessory clock sources muxing it is assumed that clocks will not be switched while running.

10.5 CKIL Synchronizing to ipg_clk

CKIL is synchronized to ipg_clk when system is in functional mode. When system is in stop mode, i.e. when there is no ipg_clk, the CKIL synchronizer will be bypassed, and raw ckil will be supplied to the system.

10.6 Low Power Clock Gating module (LPCG)

The LPCG module receives the root clocks and splits them to clock branches for each module. The clock branches are gated clocks. The enables for those gates can come from the following sources:

- Clock enable signal from CCM — this signal is generated by configuration of the CCM_CCGR bits in CCM. It is based on the low power mode.
- Clock enable signal from the module — this signal is generated by the module based on internal logic of the module. Not every enable signal from the module is used. For clock enable signals from the module, that are used, CCM will generate override signal based on programable bit in CCM_CMEOR.
- Clock enable signal from the System Reset Controller (SRC) — this signal will enable the clock during the reset procedure. See System Reset Controller(SRC) for details on the clock enable signal during reset procedure.
- Hard coded enable from fuse box.

The above possible enable signals are ANDed to generate the enable signal for the gating cell.

The enable signal for the gating cell is synchronized with the clock it needs to gate. This is done in order to prevent glitches on the gated clock.

Notifications are generated for the CCM, to indicate when to close and to open the clock roots. All notifications that correspond to the same clock root will be ORed to generate one notification signal to CCM for clock root gating.

The following figure describes the implementation for each gating cell:

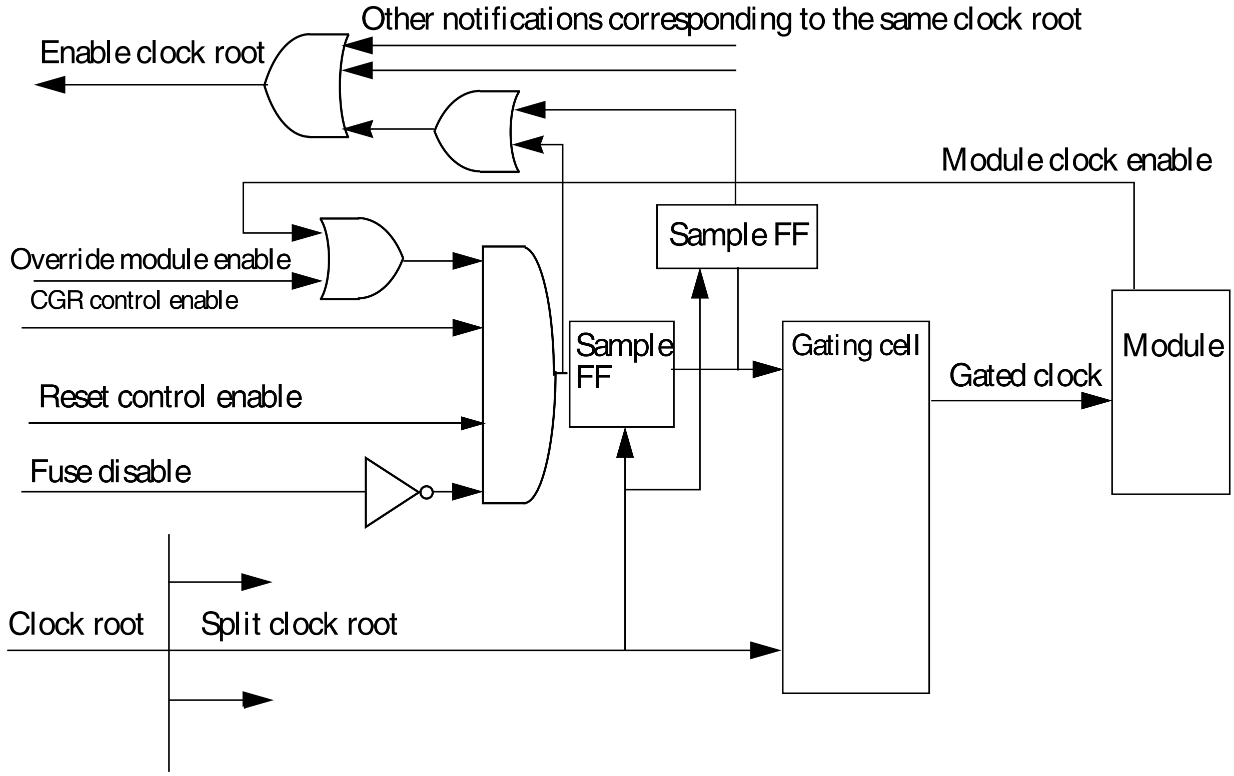


Figure 10-45. Implementation for each gating cell

The following figure describes the clock split inside LPCG module. It describes the case of two modules, one module without enable signal and one with enable signal. (SRC enable signals and sync FFs are omitted from this figure).

10.7 Power modes

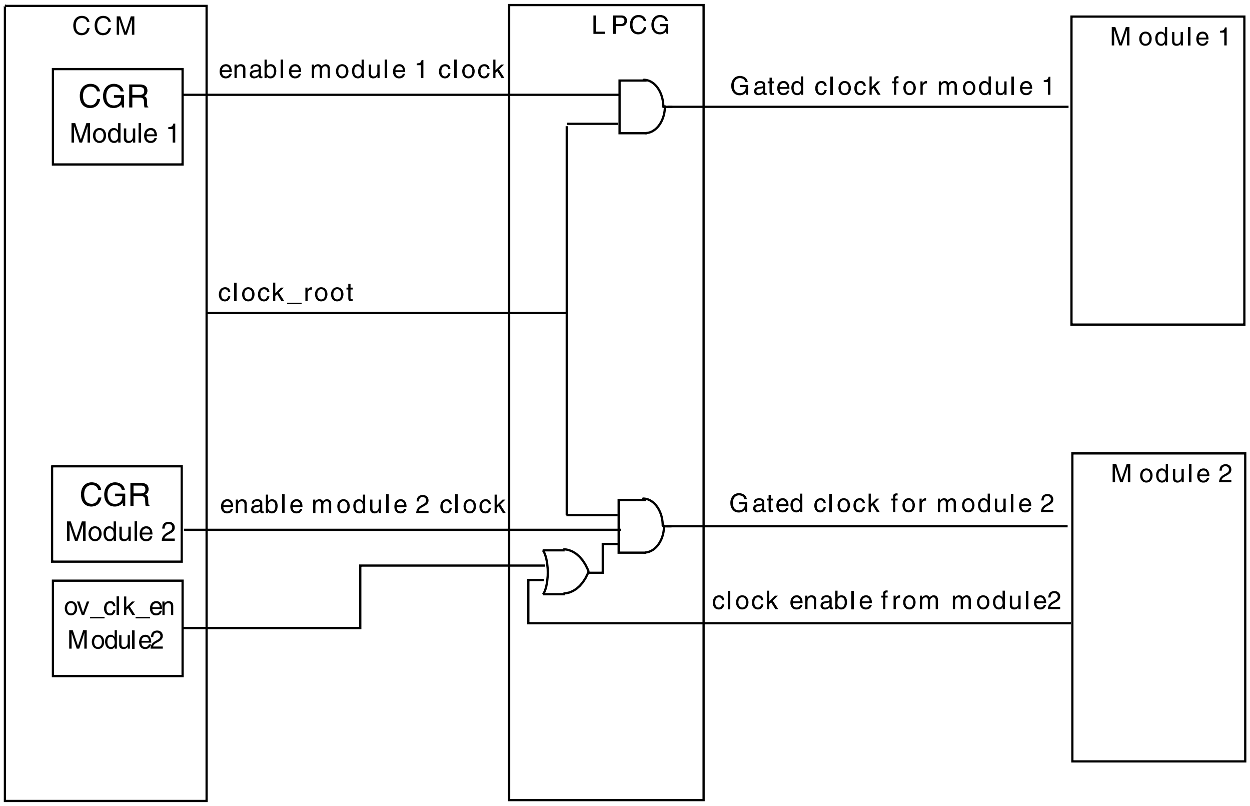


Figure 10-46. Power Modes

CCM supports three low power modes: RUN mode, WAIT mode, and STOP mode.

Chapter 11
Analog components control digital interface (ANADIG)

11.1 Overview

ANADIG is collection of digital interfaces and controllers of analog components, which include control registers described below for controlling analog components and analog interfaces. ANADIG can be accessed through the IPS bus interface.

NOTE

For the ANADIG PLL (including USB PLL), regulator, and miscellaneous registers, the fields are preprogrammed for a configuration and are read-only from a customer perspective.

NOTE

The registers described in this chapter will be referred to from different sections of this Reference Manual.

NOTE

The device has to run on the IRC to detect a BO event on the LDO1p1.

11.2 Memory Map and Registers

11.2.1 Anadig Registers

ANADIG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_0010	PLL3 Control register (ANADIG_PLL3_CTRL)	32	R/W	0001_2000h	11.21.1/ 692
4005_0020	PLL7 Control register (ANADIG_PLL7_CTRL)	32	R/W	0001_2000h	11.21.2/ 694
4005_0030	PLL2 Control register (ANADIG_PLL2_CTRL)	32	R/W	0001_3001h	11.21.3/ 696
4005_0040	PLL2 Spread Spectrum definition register (ANADIG_PLL2_SS)	32	R/W	0000_0000h	11.21.4/ 698
4005_0050	PLL2 Numerator definition register (ANADIG_PLL2_NUM)	32	R/W	0000_0000h	11.21.5/ 699
4005_0060	PLL2 Denominator definition register (ANADIG_PLL2_DENOM)	32	R/W	0000_0012h	11.21.6/ 699
4005_0070	PLL4 Control register (ANADIG_PLL4_CTRL)	32	R/W	0001_1031h	11.21.7/ 700
4005_0080	PLL4 Numerator register (ANADIG_PLL4_NUM)	32	R/W	04DD_2F15h	11.21.8/ 701
4005_0090	PLL4 Denominator register (ANADIG_PLL4_DENOM)	32	R/W	1FFF_FFDBh	11.21.9/ 702
4005_00A0	PLL6 Control register (ANADIG_PLL6_CTRL)	32	R/W	0001_1028h	11.21.10/ 703
4005_00B0	PLL6 Numerator register (ANADIG_PLL6_NUM)	32	R/W	0000_0000h	11.21.11/ 704
4005_00C0	PLL6 Denominator register (ANADIG_PLL6_DENOM)	32	R/W	0000_0012h	11.21.12/ 705
4005_00E0	PLL5 Control register (ANADIG_PLL5_CTRL)	32	R/W	0001_1001h	11.21.13/ 706
4005_00F0	ANADIG PLL3 PFD definition register (ANADIG_PLL3_PFD)	32	R/W	1B1D_1A1Ch	11.21.14/ 707
4005_0100	ANADIG PLL2 PFD definition register (ANADIG_PLL2_PFD)	32	R/W	171C_1813h	11.21.15/ 709
4005_0110	ANADIG Regulator 1P1 definition register (ANADIG_REG_1P1)	32	R/W	0002_0003h	11.21.16/ 712
4005_0120	ANADIG Regulator 3P0 definition register (ANADIG_REG_3P0)	32	R/W	0000_0004h	11.21.17/ 714
4005_0130	ANADIG Regulator 2P5 definition register (ANADIG_REG_2P5)	32	R/W	0002_0003h	11.21.18/ 716

Table continues on the next page...

ANADIG memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_0150	ANADIG Analog Miscellaneous definition register (ANADIG_ANA_MISC0)	32	R/W	0400_0000h	11.21.19/ 718
4005_0160	ANADIG Analog Miscellaneous definition register (ANADIG_ANA_MISC1)	32	R/W	0000_0000h	11.21.20/ 720
4005_0260	ANADIG Digital Program register (ANADIG_ANADIG_DIGPROG)	32	R/W	0063_0000h	11.21.21/ 721
4005_0270	PLL1 Control register (ANADIG_PLL1_CTRL)	32	R/W	0001_3001h	11.21.22/ 722
4005_0280	PLL1 Spread Spectrum register (ANADIG_PLL1_SS)	32	R/W	0000_0000h	11.21.23/ 724
4005_0290	PLL1 Numerator register (ANADIG_PLL1_NUM)	32	R/W	0000_0000h	11.21.24/ 725
4005_02A0	PLL1 Denominator register (ANADIG_PLL1_DENOM)	32	R/W	0000_0012h	11.21.25/ 725
4005_02B0	ANADIG PLL1_PFD definition register (ANADIG_PLL1_PFD)	32	R/W	See section	11.21.26/ 726
4005_02C0	ANADIG PLL Lock register (ANADIG_PLL_LOCK)	32	R/W	0000_0000h	11.21.27/ 728

11.21.1 PLL3 Control register (ANADIG_PLL3_CTRL)

This register defines the control bits for 480 MHz PLL of USB0.

Address: 4005_0000h base + 10h offset = 4005_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	0														BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BYPASS_CLK_SRC	ENABLE	POWER	0					EN_USB_CLKS	0	0			DIV_SELECT	Reserved
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_PLL3_CTRL field descriptions

Field	Description
31 LOCK	Lock bit. 0 PLL is not currently locked. 1 PLL is currently locked.
30–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ANADIG_PLL3_CTRL field descriptions (continued)

Field	Description
16 BYPASS	Bypass the PLL. 0 Disable bypass 1 Enable bypass
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 BYPASS_CLK_ SRC	Bypass Clock Source selection. 0 24M XTAL clock is selected as clock source for the PLL. 1 External clock through LVDS pad is selected as clock source for the PLL.
13 ENABLE	Enables the PLL output clock. 0 PLL output clock is gated, so disabled. 1 PLL output clock is enabled.
12 POWER	Powers up the USB0 PLL. 0 Not powered up 1 Powered up
11–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 EN_USB_CLKS	Clock Gating for 8 Phase clock of USB0 PHY. 0 8-phase PLL outputs for USB0 PHY are powered down. 1 8-phase PLL outputs for USB0 PHY are powered up.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DIV_SELECT	Select PLL multiplication factor (MFI), Fout = Fref * 20 to generates Fout = 480 MHz when Fref = 24 MHz. 0 Fout = Fref*20 (default value) 1 Fout = Fref*22
0 Reserved	This field is reserved.

11.21.2 PLL7 Control register (ANADIG_PLL7_CTRL)

This register defines the control bits for the 480 MHz PLL of USB1.

Address: 4005_0000h base + 20h offset = 4005_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	0														BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BYPASS_CLK_SRC	ENABLE	POWER	Reserved					EN_USB_CLKS	0	0			DIV_SELECT	Reserved
W																
Reset	0		1	0	0	0	0	0	0	0		0	0	0	0	0

ANADIG_PLL7_CTRL field descriptions

Field	Description
31 LOCK	Lock bit. 0 PLL is not currently locked. 1 PLL is currently locked.
30–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ANADIG_PLL7_CTRL field descriptions (continued)

Field	Description
16 BYPASS	Bypasses the PLL. 0 Disable bypass 1 Enable bypass
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 BYPASS_CLK_ SRC	Bypass Clock Source selection. 0 24M XTAL clock is selected as clock source for the PLL. 1 External clock through LVDS pad is selected as clock source for the PLL.
13 ENABLE	Enables the clock output. 0 Clock output not enabled 1 Enable clock output
12 POWER	Powers up the USB1 PLL 0 Not powered up 1 Powered up
11–7 Reserved	This field is reserved.
6 EN_USB_CLKS	Clock Gating for 8 Phase clock of USB0 PHY. 0 8-phase PLL outputs for USB1 PHY are powered down. 1 8-phase PLL outputs for USB1 PHY are powered up.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DIV_SELECT	Select PLL multiplication factor (MFI), Fout = Fref * 20 to generates Fout = 480 MHz when Fref = 24 MHz. 0 Fout = Fref * 20 (default value) 1 Fout = Fref * 22
0 Reserved	This field is reserved.

11.21.3 PLL2 Control register (ANADIG_PLL2_CTRL)

This register defines the control bits for the 528 MHz PLL.

Address: 4005_0000h base + 30h offset = 4005_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	0												PFD_OFFSET_EN	DITHER_ENABLE	BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BYPASS_CLK_SRC	ENABLE	POWERDOWN	Reserved					0					DIV_SELECT	0
W																
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1

ANADIG_PLL2_CTRL field descriptions

Field	Description
31 LOCK	Lock bit. Shows whether the PLL is locked on the required frequency. 0 PLL is not currently locked. 1 PLL is currently locked.

Table continues on the next page...

ANADIG_PLL2_CTRL field descriptions (continued)

Field	Description
30–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 PFD_OFFSET_ EN	Enables an offset in the phase frequency detector. 0 Offset in the phase frequency detector is not enabled 1 Enable an offset in the phase frequency detector
17 DITHER_ ENABLE	Enables dither in the fractional modulator calculation. 0 Dither in the fractional modulator calculation is not enabled 1 Enable dither in the fractional modulator calculation.
16 BYPASS	Bypasses the PLL The frequency that is at the input of the PLL circumvents the PLL and is routed directly to the output of the PLL, i.e., the PLL has no affect on the frequency. If this bit is set, the outplut will be either 24MHz xtal clock or external clock through LVDS pad depending on the BYPASS_CLK_SOURCE bit. 0 Disable bypass 1 Enable bypass
15 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
14 BYPASS_CLK_ SRC	Bypass Clock Source selection. 0 24M XTAL clock is selected as clock source for the PLL 1 External clock through LVDS pad is selected as clock source for the PLL.
13 ENABLE	Enables the PLL output clock. 0 PLL output clock is gated, so disabled. 1 PLL output clock is enabled.
12 POWERDOWN	Powers down the PLL. 0 PLL is not powered down 1 Power down the PLL
11–7 Reserved	This field is reserved.
6–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DIV_SELECT	Frequency multipler selection (MFI). 0 Fout = Fref * 20 1 Fout = Fref * 22. If PLL frequency is to be 528MHz, Fout = 528 MHz = 24 MHz xtal * 22.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.21.4 PLL2 Spread Spectrum definition register (ANADIG_PLL2_SS)

This register defines the control bits for the 528 MHz PLL.

Address: 4005_0000h base + 40h offset = 4005_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	STOP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ENABLE	STEP														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_PLL2_SS field descriptions

Field	Description
31–16 STOP	STOP and STEP together control the modulation depth (maximum frequency change) and Modulation Depth in the SSCG mode (as per given formula). Modulation Depth = (STOP / MFD) * Fref), where MFD is the MFD field value in PLL2 Denominator definition register (ANADIG_PLL2_DENOM) register. Recommended: Modulation Depth = 2%, Modulation Frequency = 32K Modulation Frequency = (STEP / (2 * STOP)) * Fref, where Fref = 24MHz..
15 ENABLE	This bit enables the spread spectrum modulation. 0 Spectrum modulation is not enabled 1 Enable spectrum modulation
14–0 STEP	STOP and STEP together control the modulation depth (maximum frequency change) and Modulation Depth in the SSCG mode (as per given formula). Modulation Depth = (STOP / MFD) * Fref), where MFD is the MFD field value in PLL2 Denominator definition register (ANADIG_PLL2_DENOM) register. Recommended: Modulation Depth = 2%, Modulation Frequency = 32K Modulation Frequency = (STEP / (2 * STOP)) * Fref, where Fref = 24MHz..

11.21.5 PLL2 Numerator definition register (ANADIG_PLL2_NUM)

This register defines the control bits for the 528 MHz PLL.

Address: 4005_0000h base + 50h offset = 4005_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		MFN																													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_PLL2_NUM field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–0 MFN	30-bit numerator of the fractional loop divider (unsigned integer). **NOTE:** These bits are preset to required values that are within the range of the PLL. **NOTE:** Zero is an invalid value. **NOTE:** The value of the numerator must always be configured to be less than the value of the denominator.

11.21.6 PLL2 Denominator definition register (ANADIG_PLL2_DENOM)

This register defines the control bits for the 528 MHz PLL.

Address: 4005_0000h base + 60h offset = 4005_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		MFD																													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

ANADIG_PLL2_DENOM field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–0 MFD	30-bit denominator of the fractional loop divider (unsigned integer). **NOTE:** These bits are preset to required values that are within the range of the PLL. **NOTE:** Zero is an invalid value. **NOTE:** The value of the numerator must always be configured to be less than the value of the denominator.

11.21.7 PLL4 Control register (ANADIG_PLL4_CTRL)

This register defines the control bits for the audio PLL.

Address: 4005_0000h base + 70h offset = 4005_0070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	0										0		PFD_OFFSET_EN	DITHER_ENABLE	BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BYPASS_CLK_SRC	ENABLE	POWERDOWN	0					DIV_SELECT						
W																
Reset	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0	1

ANADIG_PLL4_CTRL field descriptions

Field	Description
31 LOCK	Lock bit. 0 PLL is not currently locked. 1 PLL is currently locked.
30–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ANADIG_PLL4_CTRL field descriptions (continued)

Field	Description
20–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 PFD_OFFSET_ EN	Enables an offset in the phase frequency detector. 0 PFD offset is not enabled 1 Enable PFD offset
17 DITHER_ ENABLE	Enables dither in the fractional modulator calculation. 0 Dither is not enabled 1 Enable ditther
16 BYPASS	Bypasses the PLL. 0 No Bypass, PLL will drive its own Clock 1 Enable bypass
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 BYPASS_CLK_ SRC	Bypass Colock Source selection. 0 24M XTAL clock is selected as clock source for the PLL. 1 External clock through LVDS pad is selected as clock source for the PLL.
13 ENABLE	Enables the clock output. 0 PLL output clock is disabled. 1 PLL output clock is enabled.
12 POWERDOWN	Powers down the PLL. 0 No Power Down 1 PLL Power Down
11–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 DIV_SELECT	Frequency multipler factor selection. Freq = InputFreq (24MHz for Xtal) * DIV_SELECT **NOTE:** This PLL can function from 600MHz to 1300MHz.. **NOTE:** Zero is an invalid value.

11.21.8 PLL4 Numerator register (ANADIG_PLL4_NUM)

This register defines the control bits for the audio PLL.

Address: 4005_0000h base + 80h offset = 4005_0080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		MFN																													
W																																
Reset	0	0	0	0	0	1	0	0	1	1	0	1	1	1	0	1	0	0	1	0	1	1	1	1	0	0	0	1	0	1	0	1

ANADIG_PLL4_NUM field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–0 MFN	This is the 30-bit numerator of the fractional loop divider. **NOTE:** The value of the numerator must always be configured to be less than the value of the denominator.

11.21.9 PLL4 Denominator register (ANADIG_PLL4_DENOM)

This register defines the control bits for the audio PLL.

Address: 4005_0000h base + 90h offset = 4005_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		MFD																													
W																																
Reset	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1

ANADIG_PLL4_DENOM field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–0 MFD	This is the 30-bit denominator of the fractional loop divider. **NOTE:** The value of the numerator must always be configured to be less than the value of the denominator.

11.21.10 PLL6 Control register (ANADIG_PLL6_CTRL)

This register defines the control bits for the video PLL.

Address: 4005_0000h base + A0h offset = 4005_00A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	0										0		PFD_OFFSET_EN	DITHER_ENABLE	BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BYPASS_CLK_SRC	ENABLE	POWERDOWN	0					DIV_SELECT						
W																
Reset	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0

ANADIG_PLL6_CTRL field descriptions

Field	Description
31 LOCK	Lock bit. 0 PLL is not currently locked. 1 PLL is currently locked.
30–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ANADIG_PLL6_CTRL field descriptions (continued)

Field	Description
20–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 PFD_OFFSET_ EN	Enables an offset in the phase frequency detector. 0 PFD offset is not enabled 1 Enable PFD offset
17 DITHER_ ENABLE	Enables dither in the fractional modulator calculation. 0 Dither is not enabled 1 Enable dither
16 BYPASS	Bypasses the PLL. 0 Bypass is not enabled 1 Enable bypass
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 BYPASS_CLK_ SRC	Bypass Clock Source selection. 0 24M XTAL clock is selected as clock source for the PLL. 1 External clock through LVDS pad is selected as clock source for the PLL.
13 ENABLE	Enables the clock output. 0 Clock output is not enabled 1 Enable clock output
12 POWERDOWN	Powers down the PLL. 0 Power down is not enabled 1 Enable power down
11–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 DIV_SELECT	Frequency multiplier factor selection. Freq = InputFreq (24MHz for Xtal) * DIV_SELECT. This PLL can function from 600MHz to 1300MHz. **NOTE:** Zero is an invalid value.

11.21.11 PLL6 Numerator register (ANADIG_PLL6_NUM)

This register defines the control bits for the video PLL.

Address: 4005_0000h base + B0h offset = 4005_00B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		MFN																													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_PLL6_NUM field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–0 MFN	This is the 30-bit numerator of the fractional loop divider. **NOTE:** The value of the numerator must always be configured to be less than the value of the denominator.

11.21.12 PLL6 Denominator register (ANADIG_PLL6_DENOM)

This register defines the control bits for the video PLL.

Address: 4005_0000h base + C0h offset = 4005_00C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		MFD																													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

ANADIG_PLL6_DENOM field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–0 MFD	This is the 30-bit denominator of the fractional loop divider. **NOTE:** The value of the numerator must always be configured to be less than the value of the denominator.

11.21.13 PLL5 Control register (ANADIG_PLL5_CTRL)

This register defines the control bits for the ENET PLL. The raw VCO output frequency is 500 MHz. This PLL produces a reference clock for the Ethernet block. Ref_Ethernet is configurable based on the DIV_SELECT field.

Address: 4005_0000h base + E0h offset = 4005_00E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	0												PFD_OFFSET_EN	DITHER_ENABLE	BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BYPASS_CLK_SRC	ENABLE	POWERDOWN	0					0					DIV_SELECT	
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

ANADIG_PLL5_CTRL field descriptions

Field	Description
31 LOCK	Lock bit.

Table continues on the next page...

ANADIG_PLL5_CTRL field descriptions (continued)

Field	Description
	0 PLL is not currently locked. 1 PLL is currently locked.
30–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 PFD_OFFSET_EN	Enables an offset in the phase frequency detector. 0 PFD offset is not enabled 1 Enable PFD offset
17 DITHER_ENABLE	Enables dither in the fractional modulator calculation. 0 Dither is not enabled 1 Enable dither
16 BYPASS	Bypasses the PLL.. 0 Bypass is not enabled 1 Enable bypass
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 BYPASS_CLK_SRC	Bypass Clock Source selection. 0 24M XTAL clock is selected as clock source for the PLL. 1 External clock through LVDS pad is selected as clock source for the PLL.
13 ENABLE	Enables the clock output. 0 Clock output is not enabled 1 Enable clock output
12 POWERDOWN	Powers down the PLL. 0 Power down is not enabled 1 Enable power down
11–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 DIV_SELECT	Controls the frequency of the Ethernet reference clock. 01 50 MHz

11.21.14 ANADIG PLL3 PFD definition register (ANADIG_PLL3_PFD)

This register defines the control bits for the PFD clocks derived from PLL3 for USB0.

NOTE

It is recommended that PFD setting are kept between 18 and 35 for all PFDs.

NOTE

PLL7 for USB1 does not have PFDs.

Address: 4005_0000h base + F0h offset = 4005_00F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PFD4_CLKGATE	PFD4_STABLE	PFD4_FRAC						PFD3_CLKGATE	PFD3_STABLE	PFD3_FRAC					
W																
Reset	0	0	0	1	1	0	1	1	0	0	0	1	1	1	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PFD2_CLKGATE	PFD2_STABLE	PFD2_FRAC						PFD1_CLKGATE	PFD1_STABLE	PFD1_FRAC					
W																
Reset	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	0

ANADIG_PLL3_PFD field descriptions

Field	Description
31 PFD4_CLKGATE	This bit controls the generation of PFD4.

Table continues on the next page...

ANADIG_PLL3_PFD field descriptions (continued)

Field	Description
	0 ref_pfd4 fractional divider clock is enabled. 1 ref_pfd4 fractional divider clock is disabled for power savings.
30 PFD4_STABLE	This read-only field is for diagnostic purpose only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
29–24 PFD4_FRAC	This field controls the fractional divide value. The resulting frequency is 480*18/pfd4_frac where the value of pfd4_frac ranges between 12 and 35.
23 PFD3_CLKGATE	This bit controls the generation of PFD3. 0 ref_pfd3 fractional divider clock is enabled. 1 ref_pfd3 fractional divider clock is disabled for power savings.
22 PFD3_STABLE	This read-only bitfield is for diagnostic purposes only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
21–16 PFD3_FRAC	This field controls the fractional divide value. The resulting frequency shall be 480*18/pfd3_frac where the value of pfd3_frac ranges from 12 to 35.
15 PFD2_CLKGATE	This bit controls the generation of PFD2. 0 ref_pfd2 fractional divider clock is enabled. 1 ref_pfd2 fractional divider clock is disabled for power savings.
14 PFD2_STABLE	This read-only bitfield is for diagnostic purposes only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
13–8 PFD2_FRAC	This field controls the fractional divide value. The resulting frequency is 480*18/pfd2_frac where the value of pfd2_frac ranges from 12 to 35.
7 PFD1_CLKGATE	This bit controls the generation of PFD1. 0 ref_pfd1 fractional divider clock is enabled. 1 ref_pfd1 fractional divider clock is disabled for power saving.
6 PFD1_STABLE	This read-only bitfield is for diagnostic purposes only since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
5–0 PFD1_FRAC	This field controls the fractional divide value. The resulting frequency shall be 480 * 18 / pfd1_frac, where the value of pfd1_frac ranges from 12 to 35. **NOTE:** It is recommended that PFD setting is kept between 18 and 35 for all PFDs.

11.21.15 ANADIG PLL2 PFD definition register (ANADIG_PLL2_PFD)

This register defines the control bits for the core PLL.

NOTE

It is recommended that PFD setting is kept between 18 and 35 for all PFDs.

Address: 4005_0000h base + 100h offset = 4005_0100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PFD4_CLKGATE	PFD4_STABLE	PFD4_FRAC						PFD3_CLKGATE	PFD3_STABLE	PFD3_FRAC					
W																
Reset	0	0	0	1	0	1	1	1	0	0	0	1	1	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PFD2_CLKGATE	PFD2_STABLE	PFD2_FRAC						PFD1_CLKGATE	PFD1_STABLE	PFD1_FRAC					
W																
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	1

ANADIG_PLL2_PFD field descriptions

Field	Description
31 PFD4_CLKGATE	This bit controls the generation of PFD4.

Table continues on the next page...

ANADIG_PLL2_PFD field descriptions (continued)

Field	Description
	0 ref_pfd4 fractional divider clock is enabled. 1 ref_pfd4 fractional divider clock is disabled for power savings.
30 PFD4_STABLE	This read-only field is for diagnostic purpose only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
29–24 PFD4_FRAC	This field controls the fractional divide value. The resulting frequency is 528*18/pfd4_frac where the value of pfd4_frac ranges between 12 and 35.
23 PFD3_CLKGATE	This bit controls the generation of PFD3. 0 ref_pfd3 fractional divider clock is enabled. 1 ref_pfd3 fractional divider clock is disabled for power savings.
22 PFD3_STABLE	This read-only bitfield is for diagnostic purposes only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
21–16 PFD3_FRAC	This field controls the fractional divide value. The resulting frequency shall be 528*18/pfd3_frac where the value of pfd3_frac ranges from 12 to 35.
15 PFD2_CLKGATE	This bit controls the generation of PFD2. 0 ref_pfd2 fractional divider clock is enabled. 1 ref_pfd2 fractional divider clock is disabled for power savings.
14 PFD2_STABLE	This read-only bitfield is for diagnostic purposes only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
13–8 PFD2_FRAC	This field controls the fractional divide value. The resulting frequency is 528*18/pfd2_frac where the value of pfd2_frac ranges from 12 to 35.
7 PFD1_CLKGATE	This bit controls the generation of PFD1. 0 ref_pfd1 fractional divider clock is enabled. 1 ref_pfd1 fractional divider clock is disabled for power saving.
6 PFD1_STABLE	This read-only bitfield is for diagnostic purposes only since the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
5–0 PFD1_FRAC	This field controls the fractional divide value. The resulting frequency shall be 528*18/pfd1_frac where the value of pfd1_frac ranges from 12 to 35. **NOTE:** It is recommended that PFD setting is kept between 18 and 35 for all PFDs.

11.21.16 ANADIG Regulator 1P1 definition register (ANADIG_REG_1P1)

This register defines the control and status bits for the 1.1 V regulator. This regulator is designed to power the digital portions of the analog cells.

Address: 4005_0000h base + 110h offset = 4005_0110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														OK_VDD1P1	BO_VDD1P1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ENABLE_PULLDOWN	ENABLE_ILIMIT	ENABLE_BO	ENABLE_LINREG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	

ANADIG_REG_1P1 field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ANADIG_REG_1P1 field descriptions (continued)

Field	Description
17 OK_VDD1P1	This status bit signals that the regulator output is ok. 0 Regulator output is not ok. 1 Regulator output is ok.
16 BO_VDD1P1	This is the status bit that shows if the regulator Brown-out is asserted of not. Brown out is the point when regulator is unable to hold its output above the configured voltage. 0 Brown-out is not detected on the regulator output. 1 Brown-out is detected on the regulator output.
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ENABLE_ PULLDOWN	Control bit to enable the pull-down circuitry in the regulator. 0 Pull-down circuitry in the regulator is not enabled 1 enable the pull-down circuitry in the regulator.
2 ENABLE_ILIMIT	Control bit to enable the current-limit circuitry in the regulator input power. Once, this bit is asserted, regulator will not sync more current then its limit. 0 Current-limit circuitry in the regulator is not enabled 1 Enable the current-limit circuitry in the regulator.
1 ENABLE_BO	Control bit to enable the brown-out circuitry in the regulator. 0 Brown-out circuitry in the regulator is not enabled 1 Enable the brown-out circuitry in the regulator.
0 ENABLE_ LINREG	Control bit to enable the regulator output. 0 Regulator output is not enabled 1 Enable the regulator output.

11.21.17 ANADIG Regulator 3P0 definition register (ANADIG_REG_3P0)

This register defines the control and status bits for the 3.0 V regulator powered by the host USB V_{BUS} pin.

Address: 4005_0000h base + 120h offset = 4005_0120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														OK_VDD3P0	BO_VDD3P0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								REG_3P0_VBUS_SEL	0				ENABLE_ILIMIT	ENABLE_BO	ENABLE_LINREG
W									REG_3P0_VBUS_SEL					ENABLE_ILIMIT	ENABLE_BO	ENABLE_LINREG
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

ANADIG_REG_3P0 field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 OK_VDD3P0	This status bit signals that the regulator output is ok. 0 Regulator output is not ok 1 Regulator output is ok
16 BO_VDD3P0	This is the status bit that shows if the regulator Brown-out is asserted of not. Brown out is the point when regulator is unable to hold its output above the configured voltage. 0 Brown-out is not detected on the regulator output. 1 Brown-out is detected on the regulator output.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 REG_3P0_ VBUS_SEL	If both V_{BUS} host and V_{BUS} otg are detected present, then this bit determines which source is utilized for generating reg_3p0. 0 Utilize host power. 1 Utilize OTG power.
6–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 ENABLE_ILIMIT	Control bit to enable the current-limit circuitry in the regulator input power. Once, this bit is asserted, regulator will not sync more current then its limit. 0 Current-limit circuitry in the regulator is not enabled 1 Enable the current-limit circuitry in the regulator
1 ENABLE_BO	Control bit to enable the brown-out circuitry in the regulator. 0 Brown-out circuitry in the regulator is not enabled 1 Enable the brown-out circuitry in the regulator
0 ENABLE_ LINREG	Control bit to enable the regulator output. 0 Regulator output is not enabled 1 Enable the regulator output.

11.21.18 ANADIG Regulator 2P5 definition register (ANADIG_REG_2P5)

This register defines the control and status bits for the 2.5 V regulator.

Address: 4005_0000h base + 130h offset = 4005_0130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													ENABLE_WEAK_LINREG	OK_VDD2P5	BO_VDD2P5
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ENABLE_PULLDOWN	ENABLE_ILIMIT	ENABLE_BO	ENABLE_LINREG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

ANADIG_REG_2P5 field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 ENABLE_ WEAK_LINREG	Enables the weak 2p5 regulator. This regulator is used when the main 2p5 regulator is disabled to keep the 2.5 V output roughly at 2.5 V. It depends directly on the value of VDDIO. The same enable is also used to enable the weak 1p1 regulator. This regulator is used when the main 1p1 regulator is disabled to keep the 1.1V output roughly at 1.1V. **NOTE:** Setting this bit is mandatory for wake-up from USB in stop mode when main regulators (1p1 and 2p5) are turned off. Weak Linreg (1p1 and 2p5) are enabled in stop mode to support wake-up from USB. 0 Weak 2p5 regulator is not enabled 1 Enable the weak 2p5 regulator
17 OK_VDD2P5	Status bit that signals when the regulator output is ok. 0 Regulator output is not ok. 1 Regulator output is ok.
16 BO_VDD2P5	This is the status bit that shows if the regulator Brown-out is asserted of not. Brown out is the point when regulator is unable to hold its output above the configured voltage. Brown-out is not detected on the regulator output. Brown-out is detected on the regulator output.
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ENABLE_ PULLDOWN	Control bit to enable the pull-down circuitry in the regulator. 0 Pull-down circuitry in the regulator is not enabled 1 Enable the pull-down circuitry in the regulator
2 ENABLE_ILIMIT	Control bit to enable the current-limit circuitry in the regulator. 0 Current-limit circuitry in the regulator is not enabled 1 Enable the current-limit circuitry in the regulator
1 ENABLE_BO	Control bit to enable the brown-out circuitry in the regulator. 0 Brown-out circuitry in the regulator is not enabled 1 Enable the brown-out circuitry in the regulator
0 ENABLE_ LINREG	Control bit to enable the regulator output. 1 Regulator output is not enabled 0 Enable the regulator output.

11.21.19 ANADIG Analog Miscellaneous definition register (ANADIG_ANA_MISC0)

This register defines the control and status bits for miscellaneous analog blocks.

Address: 4005_0000h base + 150h offset = 4005_0150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			0											OSC_XTALOK_EN	OSC_XTALOK
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		CLK_24M_IRC_XTAL_SEL	STOP_MODE_CONFIG	0				REFTOP_VBGUP	0			REFTOP_SELBIASOFF	REFTOP_LOWPOWER	REFTOP_PWDVBGUP	REFTOP_PWD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_ANA_MISC0 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 OSC_XTALOK_ EN	Oscillator Crystal Lock Enable 0 Oscillator crystal lock is not enabled 1 Enable the oscillator crystal lock
16 OSC_XTALOK	Status bit which signals that the output of the 24 MHz crystal oscillator is stable. Generated from a timer and active detection of the actual frequency. 0 Output of the 24 MHz crystal oscillator is not stable 1 Output of the 24 MHz crystal oscillator is stable
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 CLK_24M_IRC_ XTAL_SEL	Select internal 24M IRC clock or External 24M xtal clock as a source of 24MHz. 0 External 24Mhz Xtal Clock 1 24MHz Internal IRC. It is recommended to not use internal IRC for enabling PLL's.
12 STOP_MODE_ CONFIG	Configure the analog behavior in stop mode. 0 All analog except rtc powered down on stop mode assertion. 1 On the device Anatop, just the Reftop (reference bias circuit) can be kept alive in Stop mode. **NOTE:** To support wake-up from USB (Device Mode) , this bit must be set to '1' if 'analog_stop_mode' is used in stop mode (i.e if 'analog_stop_mode bit is set '1' in CCM Low Power Control Register).
11–8 Reserved	This read-only field is reserved and always has the value 0.
7 REFTOP_ VBGUP	This field signals that the analog bandgap voltage is up and stable. 0 Analog bandgap voltage is not up and stable 1 Analog bandgap voltage is up and stable
6–4 Reserved	This read-only field is reserved and always has the value 0.
3 REFTOP_ SELBIASOFF	Control bit to disable the self-bias circuit in the analog bandgap. The self-bias circuit is used by the bandgap during startup. This field should be set after the bandgap has stabilized and is necessary for best noise performance of analog blocks using the outputs of the bandgap. 0 Self-bias circuit in the analog bandgap is not diabled 1 Disable the self-bias circuit in the analog bandgap
2 REFTOP_ LOWPOWER	Control bit to enable the low-power mode in the analog bandgap. 0 Low-power mode in the analog bandgap is not enabled 1 Enable the low-power mode in the analog bandgap
1 REFTOP_ PWDVBGUP	Control bit to power-down the VBG-up detection circuitry in the analog bandgap. 0 VBG-up detection circuitry in the analog bandgap is not powered down 1 Power-down the VBG-up detection circuitry in the analog bandgap

Table continues on the next page...

ANADIG_ANA_MISC0 field descriptions (continued)

Field	Description
0 REFTOP_PWD	Control bit to power-down the analog bandgap reference circuitry. 0 Analog bandgap reference circuitry is not powered down 1 Powerdown the analog bandgap reference circuitry

11.21.20 ANADIG Analog Miscellaneous definition register (ANADIG_ANA_MISC1)

This register defines the control and status bits for miscellaneous analog blocks. The lvds1 and lvds2 controls below control the behavior of the anaclk1/1b and anaclk2/2b lvds IOs.

Address: 4005_0000h base + 160h offset = 4005_0160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	IRQ_ANA_BO	IRQ_TEMPSENSE	0												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			LVDSCLK1_IBEN	0	LVDSCLK1_OBEN	0					0				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_ANA_MISC1 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 IRQ_ANA_BO	This status bit is set to one when when any of the analog regulator brownout interrupts assert. Check the regulator status bits to discover which regulator interrupt asserted. To clear this bit write a 1 in Clear SCT mode. 0 None of the analog regulator brownout interrupts is asserted 1 Set to one when when any of the analog regulator brownout interrupts is asserted
29 IRQ_ TEMPSENSE	This status bit is set to one when when the temperature sensor interrupt asserts. To clear this bit write a 1 in Clear SCT mode.

Table continues on the next page...

ANADIG_ANA_MISC1 field descriptions (continued)

Field	Description
	0 Temperature sensor interrupt is not asserted 1 Set to one when when the temperature sensor interrupt is asserted
28–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 LVDSCLK1_ IBEN	This enables the lvds input buffer for AnaClock. Do not enable input and output buffers simutaneously. 0 LVDs input buffer not enabled for AnaClock 1 Enable the LVDs input buffer for AnaClock
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 LVDSCLK1_ OBEN	This bit enables the lvds output buffer for selected internal clock. **NOTE:** Do not enable input and output buffers simultaneously. 0 LVDs output buffer is not enabled for driving internal clock 1 Enable the LVDs output buffer for driving internal clock
9–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.21.21 ANADIG Digital Program register (ANADIG_ANADIG_DIGPROG)

Address: 4005_0000h base + 260h offset = 4005_0260h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								MAJOR																MINOR							
W																																
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_ANADIG_DIGPROG field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–8 MAJOR	Digital program major version.
7–0 MINOR	Digital program minor version.

11.21.22 PLL1 Control register (ANADIG_PLL1_CTRL)

This register defines the control bits for the 528 MHz PLL.

Address: 4005_0000h base + 270h offset = 4005_0270h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	0												PFD_OFFSET_EN	DITHER_ENABLE	BYPASS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BYPASS_CLK_SRC	ENABLE	POWERDOWN	Reserved					0					DIV_SELECT	0
W																
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1

ANADIG_PLL1_CTRL field descriptions

Field	Description
31 LOCK	Lock bit. Shows whether the PLL is locked on the required frequency. 0 PLL is not currently locked. 1 PLL is currently locked.

Table continues on the next page...

ANADIG_PLL1_CTRL field descriptions (continued)

Field	Description
30–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 PFD_OFFSET_ EN	Enables an offset in the phase frequency detector. 0 Offset in the phase frequency detector is not enabled 1 Enable an offset in the phase frequency detector
17 DITHER_ ENABLE	Enables dither in the fractional modulator calculation. 0 Dither in the fractional modulator calculation is not enabled 1 Enable dither in the fractional modulator calculation.
16 BYPASS	Bypasses the PLL The frequency that is at the input of the PLL circumvents the PLL and is routed directly to the output of the PLL, i.e., the PLL has no affect on the frequency. If this bit is set, the outplut will be either 24MHz xtal clock or external clock through LVDS pad depending on the BYPASS_CLK_SOURCE bit. 0 Disable bypass 1 Enable bypass
15 Reserved	Reserved. This field is reserved. This read-only field is reserved and always has the value 0.
14 BYPASS_CLK_ SRC	Bypass Clock Source. 0 24M XTAL clock is selected as clock source for the PLL 1 External clock through LVDS pad is selected as clock source for the PLL.
13 ENABLE	Enables the PLL output clock. 0 PLL output clock is gated, so disabled. 1 PLL output clock is enabled.
12 POWERDOWN	Powers down the PLL. 0 PLL is not powered down 1 Power down the PLL
11–7 Reserved	This field is reserved.
6–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DIV_SELECT	Frequency multipler selection. 00 Fout = Fref * 20 01 Fout = Fref * 22
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.21.23 PLL1 Spread Spectrum register (ANADIG_PLL1_SS)

This register defines the control bits for the 528 MHz PLL.

Address: 4005_0000h base + 280h offset = 4005_0280h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	STOP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ENABLE	STEP														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_PLL1_SS field descriptions

Field	Description
31–16 STOP	STOP and STEP together control the modulation depth (maximum frequency change) and Modulation Depth in the SSCG mode (as per given formula). Modulation Depth = (STOP / B) * Fref), where B is the B field value in PLL2 Denominator definition register (ANADIG_PLL2_DENOM) register. Recommended: Modulation Depth = 2%, Modulation Frequency = 32K Modulation Frequency = (STEP / (2 * STOP)) * Fref, where Fref = 24MHz..
15 ENABLE	This bit enables the spread spectrum modulation. 0 Spread spectrum modulation is not enabled 1 Enable spread spectrum modulation
14–0 STEP	STOP and STEP together control the modulation depth (maximum frequency change) and Modulation Depth in the SSCG mode (as per given formula). Modulation Depth = (STOP / B) * Fref), where B is the B field value in PLL2 Denominator definition register (ANADIG_PLL2_DENOM) register. Recommended: Modulation Depth = 2%, Modulation Frequency = 32K Modulation Frequency = (STEP / (2 * STOP)) * Fref, where Fref = 24MHz..

11.21.24 PLL1 Numerator register (ANADIG_PLL1_NUM)

This register defines the control bits for the 528 MHz PLL.

Address: 4005_0000h base + 290h offset = 4005_0290h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		MFN																													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_PLL1_NUM field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–0 MFN	This is the 30-bit numerator of the fractional loop divider. **NOTE:** The value of the numerator must always be configured to be less than the value of the denominator.

11.21.25 PLL1 Denominator register (ANADIG_PLL1_DENOM)

This register defines the control bits for the 528 MHz PLL.

Address: 4005_0000h base + 2A0h offset = 4005_02A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		MFD																													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

ANADIG_PLL1_DENOM field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–0 MFD	This is the 30-bit denominator of the fractional loop divider. **NOTE:** The value of the numerator must always be configured to be less than the value of the denominator.

11.21.26 ANADIG PLL1_PFD definition register (ANADIG_PLL1_PFD)

This register defines the control bits for the core PLL.

Address: 4005_0000h base + 2B0h offset = 4005_02B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PFD4_CLKGATE	PFD4_STABLE	PFD4_FRAC						PFD3_CLKGATE	PFD3_STABLE	PFD3_FRAC					
W																
Reset	0	x*	0	1	0	0	1	0	0	x*	0	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PFD2_CLKGATE	PFD2_STABLE	PFD2_FRAC						PFD1_CLKGATE	D1_STABLE	PFD1_FRAC					
W																
Reset	0	x*	0	1	0	1	0	1	0	x*	0	1	0	0	1	1

* Notes:
- x = Undefined at reset.

ANADIG_PLL1_PFD field descriptions

Field	Description
31 PFD4_CLKGATE	This bit enables the fractional divider clock. 0 ref_pfd4 fractional divider clock is enabled. 1 ref_pfd4 fractional divider clock is disabled for power savings.
30 PFD4_STABLE	This read-only bitfield is for diagnostic purpose only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
29–24 PFD4_FRAC	This field controls the fractional divide value. The resulting frequency will be 528*18/pfd4_frac where the value of pfd4_frac ranges from 12 to 35.
23 PFD3_CLKGATE	This bit enables the fractional divider clock. 0 ref_pfd3 fractional divider clock is enabled. 1 ref_pfd3 fractional divider clock is disabled for power savings.
22 PFD3_STABLE	This read-only bitfield is fordiagnostic purposes only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
21–16 PFD3_FRAC	This field controls the fractional divide value. The resulting frequency shall be 528*18/pfd3_frac where the value of pfd3_frac ranges from 12 to 35.
15 PFD2_CLKGATE	This bit controls the generation of PFD2. 0 ref_pfd2 fractional divider clock is enabled. 1 ref_pfd2 fractional divider clock is disabled for power savings.
14 PFD2_STABLE	This read-only bitfield is for diagnostic purposes only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
13–8 PFD2_FRAC	This field controls the fractional divide value. The resulting frequency is 480*18/pfd2_FRAC where the value of pfd2_FRAC ranges from 12 to 35.
7 PFD1_CLKGATE	This bit enables the fractional divider clock. 0 ref_pfd1 fractional divider clock is enabled. 1 ref_pfd1 fractional divider clock is disabled for power saving.
6 D1_STABLE	This read-only bitfield is for diagnostic purposes only because the fractional divider should become stable quickly enough that this field will never need to be used by either device driver or application code. The value inverts when the new programmed fractional divide value has taken effect. Read this bit, program the new value, and when this bits inverts, the phase divider clock ouput is stable. Note that the value will not invert when the fractional divider is taken out of or placed into the clock-gated state.
5–0 PFD1_FRAC	This field controls the fractional divide value. The resulting frequency shall be 528*18/pfd1_frac where the value of pfd1_frac ranges from 12 to 35.

11.21.27 ANADIG PLL Lock register (ANADIG_PLL_LOCK)

This register give lock status of different PLLs controlled by ANADIG.

Address: 4005_0000h base + 2C0h offset = 4005_02C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									PLL1	PLL2	PLL4	PLL6	PLL5	PLL3	PLL7
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ANADIG_PLL_LOCK field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 PLL1	This bit shows if PLL1 (528 SYS) is locked. 0 PLL7 not locked yet. 1 PLL7 locked.
5 PLL2	This bit shows id PLL2 (528) is locked. 0 PLL3 not locked yet. 1 528 PLL3 locked.
4 PLL4	This bit shows if PLL4 (Audio) is locked. 0 PLL5 not locked yet. 1 PLL5 locked.
3 PLL6	This bit shows if PLL6 (Video) is locked. 0 PLL6 not locked yet. 1 PLL6 locked.
2 PLL5	This bit shows if PLL5 (ENET)is locked. 0 PLL4 not locked yet. 1 PLL4 locked.
1 PLL3	This bit shows if PLL3 (USB0) is locked. 0 PLL2 not locked yet. 1 PLL2 locked.
0 PLL7	This bit shows if PLL7 (USB1) is locked. 0 PLL1 not locked yet. 1 PLL1 locked.

Chapter 12
Slow Clock Source Controller Module (SCSC)

12.1 Introduction

The Slow Clock Source Controller (SCSC) module configures the SIRC and SOSC controls. It is alive in standby mode and also in case of power failure. Power Failure implies when VDDIO (3V3) or VDD_SOC (1V2) are not available.

- SIRC- 128kHz controller:

XTAL_OK	Power Fail	SIRC CONTROL
0	0	SIRC_EN
1	0	SIRC_EN
0	1	SIRC_EN_ON_FAIL
1	1	0 - SIRC will be disabled

- RC trims coming from fuse box controller are latched to retain the value in standby.
- SOSC - 32 KHz controller
  - Controls the enable signal.
  - Controls the bypass.

12.2 Memory Map and Registers

SCSC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_2000	SIRC Control Register (SCSC_SIRC_CTR)	32	R/W	0000_FF11h	12.2.1/730
4005_2004	SOSC Control (SCSC_SOSC_CTR)	32	R/W	7CFF_0000h	12.2.2/731

12.2.1 SIRC Control Register (SCSC_SIRC_CTR)

Address: 4005_2000h base + 0h offset = 4005_2000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SIRC_DIV							0			SIRC_EN_ON_FAIL	0			SIRC_EN
W																
Reset	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	1

SCSC_SIRC_CTR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value 0.
14–8 SIRC_DIV	0000000: Divide by 1 0000001: Divide by 2 ... 1111110: Divide by 127 1111111: Divide by 128 - default **NOTE:** This divider output is used as 1 KHz clock source for LPTIMER. SIRC clock is unaffected by this divider.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 SIRC_EN_ON_FAIL	This bit will have no effect when power (VDDIO or VDDSOC) is available. **NOTE:** If power fails and xtal32 clock is available - SIRC will be disabled by hardware to save power when running on battery supply. 1 If Power Fails and xtal32 is not available, enable SIRC 0 If Power Fails and xtal32 is not available. Disable SIRC
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 SIRC_EN	This bit has no effect in case Power Fail is 1. **NOTE:** The PLL lock detection circuit depends on this clock.

Table continues on the next page...

SCSC_SIRC_CTR field descriptions (continued)

Field	Description
	1 Enable SIRC 0 Disable SIRC

12.2.2 SOSC Control (SCSC_SOSC_CTR)

Figure represents the SOSC_CTR Register (CSR).

Address: 4005_2000h base + 4h offset = 4005_2004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved	Reserved														
Reset	0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											0	0			SOSC_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SCSC_SOSC_CTR field descriptions

Field	Description
31 Reserved	This field is reserved.
30–16 Reserved	This field is reserved.
15–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 SOSC_EN	0 Disable oscillator 32 KHz XOSC 1 Enable oscillator 32 KHz XOSC

Chapter 13
Clock Monitor Unit (CMU)

13.1 Introduction

The Clock Monitor Unit (CMU), also referred to as Clock Quality Checker or Clock Fault Detector, serves three purposes:

- Measures the frequency of clock source CLKMT0_RMN with CLKMN0_RMT as the reference clock
- Monitors CLKMN0_RMT frequency with CLKMT0_RMN as reference clock
- Monitors CLKMN1 frequency with CLKMT0_RMN as reference clock and detects if the monitored clock frequency leaves an upper or lower frequency boundary

One of the tasks is to supervise the integrity of the various clock sources on the chip, for example CLKMN0_RMT or CLKMN1. If the monitored clock frequency is less than the reference clock, or it violates an upper or lower frequency boundary, the CMU detects and reports this event.

The CMU can monitor CLKMN0_RMT, which must have a frequency higher than that of CLKMT0_RMN divided by the factor shown in CMU_CSR[RCDIV], and reports this event. The CMU can also monitor CLKMN1 and generate an event if CLKMN1 is greater than a high frequency boundary or less than a low frequency boundary. The upper and lower frequency boundaries are defined by the CMU High Frequency Reference Register (CMU_HFREFR) and CMU Low Frequency Reference Register (CMU_LFREFR).

The second task of the CMU is to provide a frequency meter, which allows measuring the frequency of a clock source against a reference clock. This is useful to allow the calibration of the metered clocks (such as, CLKMT0_RMN), as well as to be able to correct/calculate the time deviation of a counter that is clocked by the metered clocks.

Note

See the "Clocking" chapter for the number of instances of the CMU in this chip.

13.1.1 Main features

- CLKMT0_RMN frequency measurement with CLKMN0_RMT as reference clock.
- CLKMN0_RMT monitoring with respect to CLKMT0_RMN ÷ n clock.
- Upper or lower frequency boundary monitoring of CLKMN1 with respect to CLKMT0_RMN ÷ 4.
- Event generation for various failures detected inside monitoring unit.

13.2 Block diagram

The block diagram of the CMU module(s) is shown in the “Clocking" chapter of this Reference Manual.

13.3 Signals

The table below describes the signals on the boundary of the CMU (in alphabetical order).

Table 13-1. Signal description

Signal	I/O	Description
CLKMN0_RMT	I	Monitored Clock Signal 0/Metered Clock Signal Reference — Receives a clock signal that the CMU compares to a specified low-limit frequency to determine whether the frequency of the clock signal is greater than the specified limit. Also provides a reference clock signal for all metered clock signals.
CLKMN1	I	Monitored Clock Signal 1 — Receives a clock signal that the CMU compares to specified low-limit and high-limit frequencies to determine whether the frequency of the clock signal is between the specified limits.
CLKMT0_RMN	I	Metered Clock Signal 0/Monitored Clock Signal Reference — Receives a clock signal that the CMU measures against a reference clock frequency. Also provides a reference clock signal for all monitored clock signals.

NOTE

See the "Clocking" chapter for device specific clock sources of each CMU.

13.4 Register description and memory map

This section describes in address order all the CMU registers. Each description includes a standard register diagram with an associated figure number. The CMU memory map is listed in the following table.

NOTE

See "Clocking" chapter for register and field availability details.

CMU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_F000	CMU Control Status Register (CMU_CSR)	32	R/W	See section	13.4.1/735
4006_F004	CMU Frequency Display Register (CMU_FDR)	32	R	0000_0000h	13.4.2/737
4006_F008	CMU High Frequency Reference Register CLKMN1 (CMU_HFREFR)	32	R/W	0000_0FFFh	13.4.3/737
4006_F00C	CMU Low Frequency Reference Register CLKMN1 (CMU_LFREFR)	32	R/W	0000_0000h	13.4.4/738
4006_F010	CMU Interrupt Status Register (CMU_ISR)	32	w1c	See section	13.4.5/738
4006_F018	CMU Measurement Duration Register (CMU_MDR)	32	R/W	0000_0000h	13.4.6/740

13.4.1 CMU Control Status Register (CMU_CSR)

Address: 4006_F000h base + 0h offset = 4006_F000h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	0								SFM	0						
W																
Reset	0	0	0	0	0	0	0	0	0*	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0						CKSEL1		0					RCDIV		CME
W																
Reset	0	0	0	0	0	0	0*	0*	0	0	0	0	0	0*	0*	0

* Notes:
- RCDIV field: Not all CMU blocks will utilize this feature. See the “Clocking” chapter for CMU implementation details.
- CKSEL1 field: Not all CMU blocks will utilize this feature. See the “Clocking” chapter for CMU implementation details.
- SFM field: Not all CMU blocks will utilize this feature. See the “Clocking” chapter for CMU implementation details.

CMU_CSR field descriptions

Field	Description
0–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 SFM	Start frequency measure. The software can only set this bit to start a clock frequency measure. It is reset by hardware when the measure is ready in the CMU_FDR. 0 Frequency measurement is completed or not yet started 1 Frequency measurement is not completed
9–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–23 CKSEL1	Frequency measure clock selection bit. CKSEL1 selects the clock to be measured by the frequency meter. This only affects CMU instances that utilizes clock metering. Not all CMU blocks will utilize this feature. See the “Clocking” chapter for device specific CMU implementation details. 00 CLKMT0_RMN is selected 01 Reserved 10 Reserved 11 CLKMT0_RMN is selected
24–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–30 RCDIV	CLKMT0_RMN division factor. These bits specify the CLKMT0_RMN division factor. The output clock frequency is $f_{CLKMT0\_RMN} \div 2^{RCDIV}$. This output clock is used as reference clock to compare with CLKMN0_RMT for crystal clock monitor feature. 00 CLKMT0_RMN ÷ 1 (No division) 01 CLKMT0_RMN ÷ 2 10 CLKMT0_RMN ÷ 4 11 CLKMT0_RMN ÷ 8
31 CME	CLKMN1 monitor enable. 0 CLKMN1 monitor is disabled 1 CLKMN1 monitor is enabled

13.4.2 CMU Frequency Display Register (CMU_FDR)

The FDR is used to determine the measured frequency being monitored by the CMU.

Address: 4006_F000h base + 4h offset = 4006_F004h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0												FD																			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMU_FDR field descriptions

Field	Description
0–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–31 FD	Measured frequency bits. This register displays the measured frequency (f_{sel}) with respect to the reference clock (f_{CLKMN0_RMT}). The measured value is given by the following formula: $f_{sel} = (f_{CLKMN0\_RMT} \times$ CMU_MDR[MD]) ÷ CMU_FDR[FD]

13.4.3 CMU High Frequency Reference Register CLKMN1 (CMU_HFREFR)

The HFREFR is configured for the high frequency reference that the CMU will use for comparing against the monitored clock.The figure and table below show the HFREFR register.

Address: 4006_F000h base + 8h offset = 4006_F008h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																				HFREF											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

CMU_HFREFR field descriptions

Field	Description
0–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

CMU_HFREFR field descriptions (continued)

Field	Description
20–31 HFREF	High Frequency reference value. These bits determine the high reference value for the CLKMN1 frequency. The reference value is given by: (HFREF ÷ 16) × (f_{CLKMT0_RMN} ÷ 164).

13.4.4 CMU Low Frequency Reference Register CLKMN1 (CMU_LFREFR)

The LFREFR is configured for the low frequency reference that the CMU will use for comparing against the monitored clock. The figure and table below show the LFREFR register.

Address: 4006_F000h base + Ch offset = 4006_F00Ch

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0																				LFREF											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMU_LFREFR field descriptions

Field	Description
0–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–31 LFREF	Low Frequency reference value. These bits determine the low reference value for the CLKMN1 frequency. The reference value is given by: (LFREF ÷ 16) × (f_{CLKMT0_RMN} ÷ 4).

13.4.5 CMU Interrupt Status Register (CMU_ISR)

NOTE

All the flags in CMU_ISR register set asynchronously. This register must be read only after an interrupt is triggered by the CMU. Otherwise, the read access on this register may fetch an incorrect value.

NOTE

Before entering low-power stop modes, all clock frequency measurements in CMU should be disabled. Failing to do so may result in spurious interrupts.

Address: 4006_F000h base + 10h offset = 4006_F010h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	Reserved												Reserved	FHHI	FLLI	OLRI
W														w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*

* Notes:
- OLRI field: Not all CMU blocks will utilize this feature. See the "Clocking" chapter for specific CMU implementation details.

CMU_ISR field descriptions

Field	Description
0–27 Reserved	This field is reserved.
28 Reserved	This field is reserved.
29 FHHI	CLKMN1 frequency higher than high reference event status. This bit is set by hardware when CLKMN1 frequency becomes higher than HFREF value and CLKMN1 is 'ON'. It can be cleared by software by writing '1'. 0 No FHH event 1 FHH event occurred
30 FLLI	CLKMN1 frequency less than low reference event status. This bit is set by hardware when CLKMN1 frequency becomes lower than LFREF value, and CLKMN1 is 'ON'. Software clears this field by writing a '1'. 0 No FLL event 1 FLL event occurred

Table continues on the next page...

CMU_ISR field descriptions (continued)

Field	Description
31 OLRI	Oscillator frequency less than $f_{CLKMT0\_RMN} \div 2^{RCDIV}$ event status. This bit is set by hardware when the f_{CLKMN0_RMT} is less than $f_{CLKMT0\_RMN} \div 2^{RCDIV}$ frequency and CLKMN0_RMT is 'ON' . It can be cleared by software by writing '1'. 0 No OLR event 1 OLR event occurred

13.4.6 CMU Measurement Duration Register (CMU_MDR)

Address: 4006_F000h base + 18h offset = 4006_F018h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	0												MD																			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CMU_MDR field descriptions

Field	Description
0–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–31 MD	Measurement duration bits This field displays the measurement duration in terms of selected clock (CLKMT0_RMN) cycles. This value is loaded in the frequency meter down-counter. The down-counter starts counting when CMU_CSR [SFM] = 1.

13.5 Functional description

13.5.1 Frequency meter

The purpose of the frequency meter is to evaluate the deviation from the nominal metered source (such as CLKMT0_RMN) frequencies. This in turn allows either recalibration of these clocks or other timing corrections. Programming of the CMU_CSR[CKSEL1] field is used to select one of the metered clocks from a multiplexer that drives a simple Frequency Meter (see the CMU Block Diagram in the "Clocking" chapter). The reference clock for the Frequency Meter is the CLKMN0_RMT signal. The measurement starts when CMU_CSR[SFM] = 1. The measurement duration is given by the contents of CMU_MDR[MD] in terms of number of clock cycles of the selected metered clock. The CMU_CSR[SFM] bit is cleared by hardware once the frequency measurement is

complete and the count is loaded in CMU_FDR[FD]. The frequency of the selected clock (f_{sel}) can be derived from the value loaded in the CMU_FDR as shown in the following equation:

$$f_{sel} = f_{CLKMN0\_RMT} \times \frac{CMU\_MDR[MD]}{CMU\_FDR[FD]}$$

13.5.2 CLKMN0_RMT supervisor

If frequency of CLKMN0_RMT is smaller than the frequency of CLKMT0_RMN ÷ $2^{CMU\_CSR[RCDIV]}$ and CLKMN0_RMT is 'ON', then:

- The CMU writes 1 to CMU_ISR[OLRI].
- The CMU asserts the OLR signal.

Note

f_{CLKMN0_RMT} must be greater than $f_{CLKMT0\_RMN} \div 2^{CMU\_CSR[RCDIV]}$ by at least 0.5 MHz in order to guarantee correct f_{CLKMN0_RMT} monitoring.

13.5.3 CLKMN1 supervisor

The frequency of CLKMN1(f_{CLKMN1}) can be monitored by programming CMU_CSR[CME] = 1. CLKMN1 monitoring starts as soon as CMU_CSR[CME] = 1. This monitor can be disabled at any time by programming CMU_CSR[CME] = 0.

If f_{CLKMN1} is greater than the reference value determined by fields CMU_HFREFR[HFREF] and CLKMN1 is 'ON', then:

- The CMU writes 1 to CMU_ISR[FHHI].
- The CMU asserts the FHH signal.

If f_{CLKMN1} is less than a reference value determined by the bits CMU_LFREFR[LFREF] and the CLKMN1 is 'ON', then:

- The CMU writes 1 to CMU_ISR[FLLI].
- The CMU asserts the FLL signal.

Note

f_{CLKMN1} must be greater than f_{CLKMT0_RMN} ÷ 4 by at least 0.5 MHz in order to guarantee correct CLKMN1 monitoring.

Chapter 14
Power management

14.1 Introduction

The power management system on this device has the following capabilities:

- Software-controlled clock gating of peripherals through the CCM module
- Software-controlled enabling/disabling of PLL source through the Anadig registers
- Lowest-power mode features power gating and disabling of all clock sources
- Two power domains (PD0 and PD1) and two voltage domains (1.1V SNVS logic and 1.2V)
- Option to retain 64K, 16K, or no memory for different power-gated (LPStop*n*) modes
- Multiple operating modes to optimize device performance and wake-up times, including
  - Run
  - Low Power Run (LPRun, ULPRun)
  - Stop
  - Low Power Stop (LPStop1, LPStop2, LPStop)

The following figure shows the modules in the power management system and their interaction. The power domains (PD*n*) are described in Power domains.

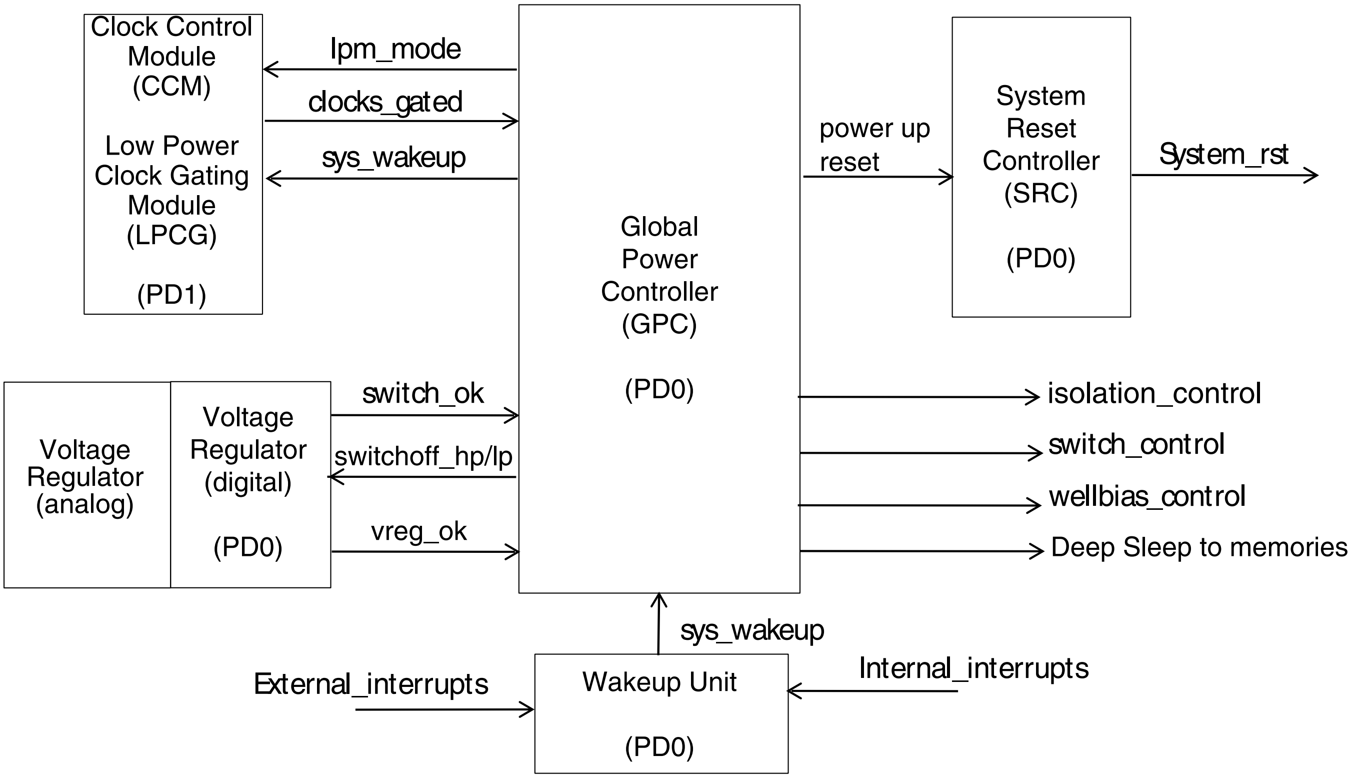


Figure 14-1. Power management modules

14.2 Power domains

The device's modules are partitioned into three logical power domains to allow the user to minimize power consumption by powering off certain functions while retaining only necessary functionality.

Table 14-1. Power domain descriptions

Power Domain	Description	Modules
PD1	Power-gated domain	All modules that are power-gated in LPStop*n* modes
PD0	Always-powered domain. Modules in this domain are operating in LPStop*n* mode.	LPTimer, SRC, VREG_DIG, GPC, WKPU, PMU, FIRC, ADC0, ADC1, LCD, 17 GPIO pads for wakeup
VBAT	The modules in this domain are operational even if there is external supply failure. This domain derives its power from the coin cell.	VBAT contains the SecureRTC, 32KHz XOSC, Tamper, and Monitors. It is powered by a coin cell when the main power supply is switched off.

14.3 Low-power modes

Warning

In order for certain low-power modes to operate as expected, unused TAMPER detection pins must be tied to ground.

The following table shows the device configuration for the various low-power modes. The table also indicates which modules are power-gated, clock-gated, or software configurable. See Power domains for details on what modules are included in the different power domains.

Table 14-2. Power gating and clock gating overview

Domain	Features	Power Gated Modes: SW save and restore[1]			LP Modes: Dynamic Power Saving and Clock Gating Modes[1]		
		LPStop1	LPStop2	LPStop3	Stop	ULPRun	LPRun
PD1	Platform	Off	Off	Off	On	On	On
PD1	Peripherals	Off	Off	Off	Off	Option	Option
PD1	PLLs	Off	Off	Off	Off	Off	Off
PD1	24M XOSC	Off	Off	Off	Off	Off	Option
PD0_SW	StbyRAM2 (64k)[2]	Off	Off	On	On	On	On
PD0_SW	StbyRAM1 (16k)[3]	Off	On	NA	On	On	On
PD0	Always-On logic	On	On	On	On	On	On
PD0	24M IRC	Off	Option	Option	Option	Option	Option
VBAT[4]	128k IRC	Option	Option	Option	Option	Option	Option
VBAT[4]	32K XOSC	Option	Option	Option	Option	Option	Option
VBAT[4]	SNVS-LP	On	On	On	On	On	On
IO[5]	IO State	Lost	Lost	Lost	Retained	Retained	Retained
PD0	HP-REG	Off	Off	Off	Option	On	On
PD0	LP-REG	Off	Off	Off	On	On	On
PD0	ULPREG	On	On	On	On	On	On
-	Vreg Wake-up	200us	200us	200us	200us	-	-

1. Option denotes that the application can configure the feature to be On or Off. Recommended configurations are given below.
2. First 64K of SysRAM0
3. First 16K of SysRAM0
4. VBAT contains the RTC, 32kHz XOSC, and Tamper. It is powered by a coin cell when the main power supply is switched off.
5. There are 17 always alive pads which can be used to cause a wakeup event in LPStop*n* modes. See the Pin muxing section for details.

14.3.1 Run mode

In Run mode:

- Device operates at highest rated frequency.
- All device peripherals are operational.

14.3.2 LPRun mode

Low Power Run mode is a software-controlled, dynamic power-saving mode. On reset exit the device is in LPRun mode. All the Arm subsystem clocks are on Fast IRC clock, which is at 24MHz. All the PLLs are in bypassed (Off) state. All the running clocks are scaled down to lower frequencies, from high PLL frequencies to FIRC or FOSC. If some auxiliary clocks, for example the audio subsystem or nand flash clocks, are to be operated at the crystal oscillator clock, then PLLs may be configured to run in bypass mode by configuring the Anadig registers.

Other features of the LPRun mode include:

- The CCM_CCGR register is configured to clock gate the peripherals.
- The PLL and PFD modules are disabled through Anadig.
- FXOSC24 is disabled through the CCM.
- FIRC24 is enabled through the CCM.

14.3.3 ULPRun mode

Ultra-Low Power Run mode is an extension of the LPRun mode allowing the Arm subsystem clock frequencies to be further scaled down to 32KHz or 128KHz.

14.3.4 Wait mode

Wait mode is entered when the GPC_LPMR register is configured for Wait mode and the WFI (Wait For Interrupt) instruction is executed on the core. Wait mode is recommended for applications where the cores can be stalled and other peripherals can operate at a lower frequency.

WFI for an individual core can be executed and GPC_LPMR may not be configured to enter into chip Wait mode. In dual-core configurations, the application can configure WFI to put a single core into halt. Other peripherals can continue operating at the configured frequencies if GPC_LPMR is configured to 00b, i.e., Run mode.

In Wait mode

- CA5 and CM4 cores are halted.
- Normal recovery is from an interrupt.

14.3.5 Stop mode

All clocks to the peripherals are gated by the CCM. Applications should configure that module if any peripherals are to be clocked in Stop mode.

14.3.5.1 Stop mode entry sequence (clock-gating mode)

The following steps must be performed to enter Stop mode.

1. Switch the system clock to 24MHz FIRC by performing the following two steps in order. This is done before disabling any of the primary clock sources (e.g., PLLs, SIRC, SOSC, FOSC) as the low-power state machine needs a running system clock.
   a. Writing 00b to CCM_CCSR[4:5]
   b. Writing 00b to CCM_CCSR[3:0]
2. Disable the PLLs by configuring these Anadig registers
   - Anadig_PLL7_CTRL
   - Anadig_PLL3_CTRL
   - Anadig_PLL2_CTRL
   - Anadig_PLL4_CTRL
   - Anadig_PLL6_CTRL
   - Anadig_PLL5_CTRL
   - Anadig_PLL1_CTRL
3. Write 1 to CCM_CLPCR[ANATOP_STOP_MODE]. This bit determines whether the PMIC_VSTBY_REQ pin (which notifies the external power management IC to switch from functional voltage to standby voltage) will be asserted in Stop mode.
4. Write 10b to GPC_LPMR[CLPCR] to enter Stop mode.
5. Execute the WFI instruction.

14.3.5.2 Optional stop mode entry sequence

This section describes the recommended configuration for advanced power-saving in Stop mode.

When the main 2p5 regulator is disabled, the weak 2p5 regulator can be enabled to keep the 2.5V output at roughly 2.5V, depending on the value of VDDIO. To use this feature, perform the following steps.

1. Write 1 to Anadig_reg_2p5[enable_linreg] to enable regulator output.
2. Write 1 to Anadig_reg_2p5[enable_weak_ linreg] to enable the weak regulator.

In addition, the GPC provides several power-saving options in Stop mode. In Stop mode you can:

- Enable Well Bias by writing 1 to GPC_PGCR[WB_ STOP].
- Turn off HPREG by writing 1 to GPC_PGCR[HP_OFF].
- Enable Deep Sleep for all memories by writing 1 to GPC_PGCR[DS_STOP]. Deep Sleep shuts down power to periphery and maintains memory contents. The outputs of the memory are pulled low.

Lowest-power Stop mode can be achieved by disabling all clock sources through the CCM and SCSC.

14.3.6 LPStop*n* modes

Low Power Stop modes are advanced power saving modes that switch off power to Power Domain 1 (PD1).

Using registers in the GPC, you can configure the device to enter into one of three LPStop modes:

- LPStop1: Lowest power stop mode with no SRAM retained
- LPStop2: 16K RAM is retained
- LPStop3: 64K RAM is retained

It is recommended to keep all the primary clock sources disabled and to wake up from external interrupt sources. Refer to Wakeup Unit (WKPU) for details. If the RTC is to be kept enabled, SOSC can be used as clock source.

14.3.6.1 LPStop*n* mode entry sequence (power-gating mode)

The following steps must be performed to enter LPStop*n* mode.

1. Switch the system clock to 24MHz FIRC by performing the following two steps in order.
    a. Writing 00b to CCM_CCSR[4:5]
    b. Writing 00b to CCM_CCSR[3:0]
2. Disable the PLLs by configuring these Anadig registers
    - Anadig_PLL7_CTRL

- Anadig_PLL3_CTRL
- Anadig_PLL2_CTRL
- Anadig_PLL4_CTRL
- Anadig_PLL6_CTRL
- Anadig_PLL5_CTRL
- Anadig_PLL1_CTRL

3. Write 1 to CCM_CLPCR[ANATOP_STOP_MODE]. This bit determines whether the PMIC_VSTBY_REQ pin (which notifies the external power management IC to switch from functional voltage to standby voltage) will be asserted in Stop mode.
4. Write 10b to GPC_LPMR[CLPCR] to enter Stop mode.
5. Write 1 to GPC_PGCR[PG_PD1] to enter into power gated mode for PD1.
6. Execute the WFI instruction.

14.3.6.2 Optional LPStop*n* mode entry sequence

The GPC provides several power-saving options in LPStop*n* mode. In LPStop*n* mode you can:

- Keep 64KB of System RAM on by writing 0 to GPC_PGCR[PG_16K] and to GPC_PGCR[PG_48K]. Clearing these two bits will cause device to enter LPStop3 mode.
- Disable FIRC during LPStop*n* by writing 0 to CCM_CCR[FIRC_EN].

14.3.7 Interrupt connectivity

This figure shows the interrupt connectivity of the device.

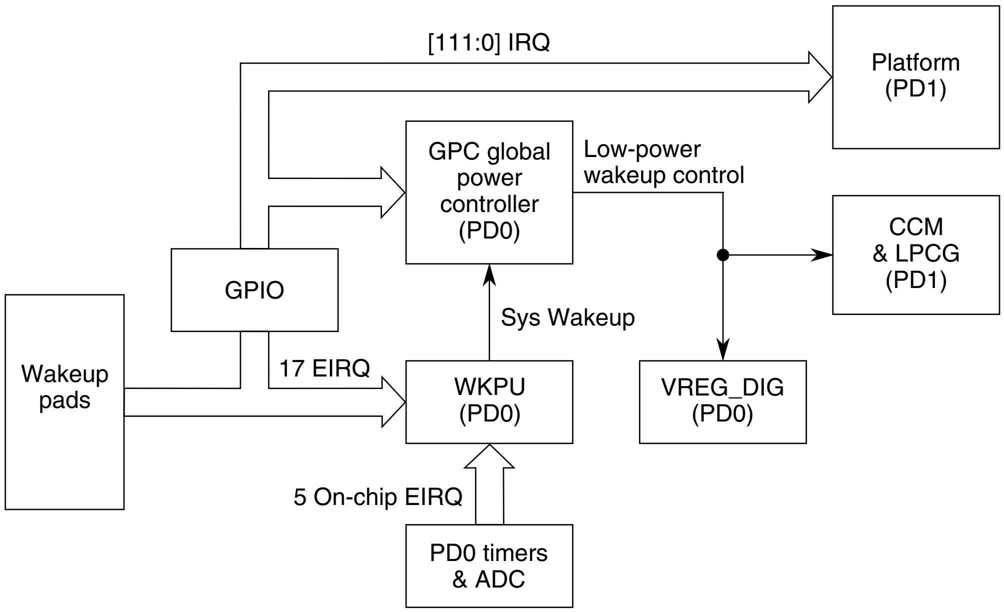


Figure 14-2. Interrupt connectivity

NOTE

- Refer to the GPC_IMR configuration registers for interrupt configurations.
- System can exit from LPStop*n* only through interrupts routed by WKPU.
- All the external IRQs should be level-triggered to cause an exit from Stop mode.
- Refer to Wakeup Unit (WKPU) for on-chip and external wakeup interrupts.

Chapter 15
Global Power Controller (GPC)

15.1 Introduction

The Global Power Controller (GPC) module generates the controls for power gated modes and low power modes. It generally does the isolation of power domain, power gating of power switches and initiates power down to the voltage regulator.

15.2 Features

Key features of the GPC include:

- Power shutdown controller
- Provides the ability to switch off power to a target subsystem.
- Generates power-up and power-down control sequences.
- Provides programmable registers to adjust the timing of the power control signals.

15.3 GPC Memory/Register Map

GPC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_C000	Power Gating Control Register (GPC_PGCR)	32	R/W	0000_00C0h	15.3.1/752
4006_C00C	Power Gating Status Register (GPC_PGSR)	32	R/W	0000_0000h	15.3.2/753
4006_C040	Low Power Mode Register (GPC_LPMR)	32	R/W	0000_0000h	15.3.3/754
4006_C044	Interrupt Mask Register 1 (GPC_IMR1)	32	R/W	0000_0000h	15.3.4/755
4006_C048	Interrupt Mask Register 2 (GPC_IMR2)	32	R/W	0000_0000h	15.3.5/757
4006_C04C	Interrupt Mask Register 3 (GPC_IMR3)	32	R/W	0000_0000h	15.3.6/760
4006_C050	Interrupt Mask Register (GPC_IMR4)	32	R/W	0000_0000h	15.3.7/763
4006_C054	Interrupt Status Register 1 (GPC_ISR1)	32	R/W	0000_0000h	15.3.8/764

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_C058	Interrupt Status Register 2 (GPC_ISR2)	32	R	0000_0000h	15.3.9/766
4006_C05C	Interrupt Status Register 3 (GPC_ISR3)	32	R	0000_0000h	15.3.10/ 768
4006_C060	Interrupt Status Register 4 (GPC_ISR4)	32	R/W	0000_0000h	15.3.11/ 771

15.3.1 Power Gating Control Register (GPC_PGCR)

The GPC_PGCR enables the response to a power-down request. It also allows to configure various power gated modes and Low Power modes. The following figure shows the register and table describes the register fields.

NOTE

Deep Sleep pin shut down power to periphery and maintain memory contents. The outputs of the memory are pulled low

Address: 4006_C000h base + 0h offset = 4006_C000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DS_STOP	DS_LPSTOP	0	WB_STOP	HP_OFF	PG_48K	PG_16K	PG_PD1
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

GPC_PGCR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 DS_STOP	0 Deep Sleep signal to memories is not asserted in STOP mode. 1 Deep Sleep signal to memories is asserted in STOP mode.
6 DS_LPSTOP	0 Deep Sleep signal to memories is not asserted in LPSTOP mode. 1 Deep Sleep signal to memories is asserted in LPSTOP mode.

Table continues on the next page...

GPC_PGCR field descriptions (continued)

Field	Description
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 WB_STOP	Enable Well Bias in STOP mode 0 Well Bias disabled in STOP mode. 1 Well Bias enabled in STOP mode.
3 HP_OFF	Turn OFF HPREG in STOP mode 0 HPREG ON in STOP mode. 1 HPREG OFF in STOP mode.
2 PG_48K	Power gate 64KRAM. This bit will take in effect if PGCR[PG_PD1] is set **NOTE:** If this bit is configured 0, PG_16K must also be configured to 0. This will enable LPSTOP3 mode 0 Do not switch off 64K RAM (16K and 48K RAM Power Switch) during LPSTOP3 mode. 1 Switch off 48K RAM Power Switch during LPSTOP2 or LPSTOP1 mode.
1 PG_16K	Power gate 16KRAM. This bit will take in effect if PGCR[PG_PD1] is set 0 Do not switch off 16K RAM Power Switch during LPSTOP3 and LPSTOP2 mode 1 Switch off 16K RAM Power Switch during LPSTOP1 mode.
0 PG_PD1	Power gate Power Domain 1. 0 Do not switch off power to power domain1. 1 Switch off power domain 1 (LPSTOP mode).

15.3.2 Power Gating Status Register (GPC_PGSR)

The GPC_PGSR provides the power-down status of the target subsystem. The following figure shows the register and table describes the register fields.

Address: 4006_C000h base + Ch offset = 4006_C00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											PREV_LPM		CUR_LPM		PSR
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PGSR field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–3 PREV_LPM	Previous Low Power Mode. These bits indicate the previous state of the GPC. 00 RUN mode 01 WAIT mode 10 STOP mode 11 LPSTOPn mode
2–1 CUR_LPM	Current Low Power Mode 00 RUN mode 01 WAIT mode 10 STOP mode 11 LPSTOPn mode
0 PSR	Power status. When in functional (or software-controlled debug) mode, PGC hardware sets PSR as soon as any of the power control output changes its state to one. Write one to clear this bit. Software should clear this bit after power up; otherwise, PSR continues to reflect the power status of the initial power down.

15.3.3 Low Power Mode Register (GPC_LPMR)

The GPC_LPMR register allows the user to configure the low power mode. The following figure shows the register and the table describes the register fields.

NOTE

To enter into LPSTOP mode, CLPCR should be configured to 2'b10 and GPC_PGCR[PG_PD1, PG_64K, PG_16K] should be configured accordingly.

Address: 4006_C000h base + 40h offset = 4006_C040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CLPCR	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_LPMR field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

GPC_LPMR field descriptions (continued)

Field	Description
1–0 CLPCR	Mode Control 00 RUN 01 WAIT 10 STOP 11 RSVD

15.3.4 Interrupt Mask Register 1 (GPC_IMR1)

This register is used to mask certain interrupts if they are not desired to be a source of wake up during STOP mode.

Address: 4006_C000h base + 44h offset = 4006_C044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DCU1	DCU0	0	SDHC1	SDHC0	DRAMC	QSPI1	QSPI0	RESERVED	RESERVED	WDOGM4	WDOGA5	0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1 field descriptions

Field	Description
31 DCU1	Masks DCU1interrupt 0 Enable DCU1 interrupt as source of wake-up from STOP mode. 1 Ignore interrupt from DCU1 as source of wake-up from STOP mode.
30 DCU0	Masks DCU0 interrupt 0 Enable DCU0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from DCU0 as source of wake-up from STOP mode.
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 SDHC1	Masks SDHC1 interrupt 0 Enable SDHC1 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from SDHC1 as source of wake-up from STOP mode.

Table continues on the next page...

GPC_IMR1 field descriptions (continued)

Field	Description
27 SDHC0	Masks SDHC0 interrupt 0 Enable SDHC0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from SDHC0 as source of wake-up from STOP mode.
26 DRAMC	Masks DRAMC interrupt 0 Enable DRAMCinterrupt as source of wake-up from STOP mode 1 Ignore interrupt from DRAMC as source of wake-up from STOP mode.
25 QSPI1	Masks QSPI1interrupt 0 Enable QSPI1 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from QSPI1 as source of wake-up from STOP mode.
24 QSPI0	Masks QSPI0 interrupt 0 Enable QSPI0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from QSPI0 as source of wake-up from STOP mode.
23 RESERVED	This field is reserved. Reserved
22 RESERVED	This field is reserved.
21 WDOGM4	Masks WDOGM4 interrupt 0 Enable WDOGM4 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from WDOGM4 as source of wake-up from STOP mode.nable
20 WDOGA5	Masks WDOGA5 interrupt 0 Enable WDOGA5 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from WDOGA5 as source of wake-up from STOP mode.
19–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

15.3.5 Interrupt Mask Register 2 (GPC_IMR2)

This register is used to mask certain interrupts if they are not desired to be a source of wake up during STOP mode.

Address: 4006_C000h base + 48h offset = 4006_C048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UART2	UART1	UART0	MLB	FLEXCAN1	FLEXCAN0	0	DAC1	DAC0	ADC1	ADC0	0	ANADIGA	ANADIGB	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		FTM3	FTM2	FTM1	FTM0	0	LPTMR	PIT	0		LCD	RLE	Reserved	0	VIU
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2 field descriptions

Field	Description
31 UART2	Masks UART2 interrupt 0 Enable UART2 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from UART2 as source of wake-up from STOP mode
30 UART1	Masks UART1 interrupt 0 Enable UART1 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from UART1 as source of wake-up from STOP mode
29 UART0	Masks UART0 interrupt 0 Enable UART0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from UART0 as source of wake-up from STOP mode
28 MLB	Masks MLB interrupt 0 Enable MLB interrupt as source of wake-up from STOP mode 1 Ignore interrupt from MLB as source of wake-up from STOP mode
27 FLEXCAN1	Masks FLEXCAN1 interrupt 0 Enable FLEXCAN1 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from FLEXCAN1 as source of wake-up from STOP mode
26 FLEXCAN0	Masks FLEXCAN0 interrupt 0 Enable FLEXCAN0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from FLEXCAN0 as source of wake-up from STOP mode

Table continues on the next page...

GPC_IMR2 field descriptions (continued)

Field	Description
25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 DAC1	Masks DAC1interrupt 0 Enable DAC1 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from DAC1 as source of wake-up from STOP mode
23 DAC0	Masks DAC0 interrupt 0 Enable DAC0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from DAC0 as source of wake-up from STOP mode
22 ADC1	Masks ADC1 interrupt 0 Enable ADC1 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from ADC1 as source of wake-up from STOP mode
21 ADC0	Masks ADC0 interrupt 0 Enable ADC0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from ADC0 as source of wake-up from STOP mode
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 ANADIGA	Masks ANADIGA interrupt 0 Enable ANADIGA interrupt as source of wake-up from STOP mode 1 Ignore interrupt from ANADIGA as source of wake-up from STOP mode
18 ANADIGB	Masks ANADIGB interrupt 0 Enable ANADIGB interrupt as source of wake-up from STOP mode 1 Ignore interrupt from ANADIGB as source of wake-up from STOP mode
17–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 FTM3	Masks FTM3 interrupt 0 Enable FTM3 interrupt as source of wake-up from STOP 1 Ignore interrupt from FTM3 as source of wake-up from STOP mode
12 FTM2	Masks FTM2 interrupt 0 Enable FTM2 interrupt as source of wake-up from STOP 1 Ignore interrupt from FTM2 as source of wake-up from STOP mode
11 FTM1	Masks FTM1 interrupt 0 Enable FTM1 interrupt as source of wake-up from STOP 1 Ignore interrupt from FTM1 as source of wake-up from STOP mode
10 FTM0	Masks FTM0 interrupt 0 Enable FTM0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from FTM0 as source of wake-up from STOP mode
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

GPC_IMR2 field descriptions (continued)

Field	Description
8 LPTMR	Masks LPTMR interrupt 0 Enable LPTMR interrupt as source of wake-up from STOP mode 1 Ignore interrupt from LPTMR as source of wake-up from STOP mode
7 PIT	Masks PIT interrupt 0 Enable PIT interrupt as source of wake-up from STOP mode 1 Ignore interrupt from PIT as source of wake-up from STOP mode
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 LCD	Masks LCD interrupt 0 Enable SEG LCD interrupt as source of wake-up from STOP mode 1 Ignore interrupt from SEG LCD as source of wake-up from STOP mode
3 RLE	Masks RLE interrupt 0 Enable RLE interrupt as source of wake-up from STOP mode 1 Ignore interrupt from RLE as source of wake-up from STOP mode
2 Reserved	This field is reserved. Reserved
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 VIU	Masks VIU interrupt 0 Enable VIU interrupt as source of wake-up from STOP mode 1 Ignore interrupt from VIU as source of wake-up from STOP mode

15.3.6 Interrupt Mask Register 3 (GPC_IMR3)

This register is used to mask certain interrupts if they are not desired to be a source of wake up during STOP mode.

Address: 4006_C000h base + 4Ch offset = 4006_C04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CCM_B	CCM_A	0	WKPU0	CMU	ASRC	SPDIF	ESAI	UNIMPLEMENTED				NFC	ESW	1588T1	1588T0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ENET1	ENET0	0	USBC1	USBC0	I2C3	I2C2	I2C1	I2C0	DSPI3	DSPI2	DSPI1	DSPI0	UART5	UART4	UART3
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3 field descriptions

Field	Description
31 CCM_B	Masks CCM_B interrupt 0 Enable CCM_B interrupt as source of wake-up from STOP mode 1 Ignore interrupt from CCM_B as source of wake-up from STOP mode
30 CCM_A	Masks CCM_A interrupt 0 Enable CCM_A interrupt as source of wake-up from STOP mode 1 Ignore interrupt from CCM_A as source of wake-up from STOP mode
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 WKPU0	Masks WKPU0 interrupt WKPU0 0 Enable WKPU0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from WKPU0 as source of wake-up from STOP mode
27 CMU	Masks CMU interrupt 0 Enable CMU interrupt as source of wake-up from STOP mode 1 Ignore interrupt from CMU as source of wake-up from STOP mode
26 ASRC	Masks ASRC interrupt 0 Enable ASRC interrupt as source of wake-up from STOP mode 1 Ignore interrupt from ASRC as source of wake-up from STOP mode

Table continues on the next page...

GPC_IMR3 field descriptions (continued)

Field	Description
25 SPDIF	Masks SPDIF interrupt 0 Enable SPDIF interrupt as source of wake-up from STOP mode 1 Ignore interrupt from SPDIFas source of wake-up from STOP mode
24 ESAI	Masks ESAI interrupt 0 Enable ESAI interrupt as source of wake-up from STOP mode 1 Ignore interrupt from ESAI as source of wake-up from STOP mode
23–20 UNIMPLEMENTED	This field is unimplemented. Read or write on this field has no effect.
19 NFC	Masks NFC interrupt 0 Enable NFC interrupt as source of wake-up from STOP mode 1 Ignore interrupt from NFC as source of wake-up from STOP mode
18 ESW	Masks ESW interrupt 0 Enable ESW interrupt as source of wake-up from STOP mode 1 Ignore interrupt from ESW as source of wake-up from STOP mode
17 1588T1	Masks 1588T1 interrupt 0 Enable 1588T1 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from 1588T1as source of wake-up from STOP mode
16 1588T0	Masks 1588T0 interrupt 0 Enable 1588T0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from 1588T0 as source of wake-up from STOP mode
15 ENET1	Masks ENET1 interrupt 0 Enable ENET1interrupt as source of wake-up from STOP mode 1 Ignore interrupt from ENET0 as source of wake-up from STOP mode
14 ENET0	Masks ENET0 interrupt 0 Enable ENET0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from ENET0 as source of wake-up from STOP mode
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 USBC1	Masks USBC1interrupt 0 Enable USBC1interrupt as source of wake-up from STOP mode 1 Ignore interrupt from USBC1as source of wake-up from STOP mode
11 USBC0	Masks USBC0 interrupt 0 Enable USBC0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from USBC0 as source of wake-up from STOP mode
10 I2C3	Masks I2C3 interrupt 0 Enable I2C3 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from I2C3 as source of wake-up from STOP mode

Table continues on the next page...

GPC_IMR3 field descriptions (continued)

Field	Description
9 I2C2	Masks I2C2 interrupt 0 Enable I2C2 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from I2C2 as source of wake-up from STOP mode
8 I2C1	Masks I2C1interrupt 0 Enable I2C1interrupt as source of wake-up from STOP mode 1 Ignore interrupt from I2C1 as source of wake-up from STOP mode
7 I2C0	Masks I2C0 interrupt 0 Enable I2C0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from I2C0 as source of wake-up from STOP mode
6 DSPI3	Masks DSPI3 interrupt 0 Enable DSPI3 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from DSPI3 as source of wake-up from STOP mode
5 DSPI2	Masks DSPI2 interrupt 0 Enable DSPI2 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from DSPI2 as source of wake-up from STOP mode
4 DSPI1	Masks DSPI1 interrupt 0 Enable DSPI1 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from DSPI1 as source of wake-up from STOP mode
3 DSPI0	Masks DSPI0 interrupt 0 Enable DSPI0 interrupt as source of wake-up from STOP mode 1 Ignore interruptfrom DSPI0 as source of wake-up from STOP mode
2 UART5	Masks UART5 interrupt 0 Enable UART5 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from UART5 as source of wake-up from STOP mode
1 UART4	Masks UART4 interrupt 0 Enable UART4 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from UART4 as source of wake-up from STOP mode
0 UART3	Masks UART3 interrupt 0 Enable UART3 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from UART3 as source of wake-up from STOP mode

15.3.7 Interrupt Mask Register (GPC_IMR4)

This register is used to mask certain interrupts if they are not desired to be a source of wake up during STOP mode.

Address: 4006_C000h base + 50h offset = 4006_C050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0	0				Reserved			0	EWM	PDB	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 GPIO4	Masks GPIO4 interrupt 0 Enable GPIO4 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from GPIO4 as source of wake-up from STOP mode
14 GPIO3	Masks GPIO3 interrupt 0 Enable GPIO3 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from GPIO3 as source of wake-up from STOP mode
13 GPIO2	Masks GPIO2 interrupt 0 Enable GPIO2 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from GPIO2 as source of wake-up from STOP mode
12 GPIO1	Masks GPIO1 interrupt 0 Enable GPIO1 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from GPIO1 as source of wake-up from STOP mode
11 GPIO0	Masks GPIO0 interrupt 0 Enable GPIO0 interrupt as source of wake-up from STOP mode 1 Ignore interrupt from GPIO0 as source of wake-up from STOP mode
10–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

GPC_IMR4 field descriptions (continued)

Field	Description
6–4 Reserved	This field is reserved.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 EWM	Masks EWM interrupt 0 Enable EWM interrupt as source of wake-up from STOP mode 1 Mask interrupt from EWM as source of wake-up from STOP mode
1 PDB	Mask PDB interrupt 0 Enable PDB interrupt as source of wake-up from STOP mode 1 Mask interrupt from PDB as source of wake-up from STOP
0 Reserved	**NOTE** Configuring this bit will not guarantee chip functionality. This field is reserved.

15.3.8 Interrupt Status Register 1 (GPC_ISR1)

This register indicates the source of interrupt causing system wake up during STOP mode.

Address: 4006_C000h base + 54h offset = 4006_C054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DCU1	DCU0	0	SDHC1	SDHC0	DRAMC	QSPI1	QSPI0	RESERVED	Reserved	WDOGM4	WDOGA5	0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR1 field descriptions

Field	Description
31 DCU1	DCU1 interrupt status 0 No interrupt from DCU1 1 Interrupt generated from DCU1

Table continues on the next page...

GPC_ISR1 field descriptions (continued)

Field	Description
30 DCU0	DCU0 interrupt status 0 No interrupt from DCU0 1 Interrupt generated from DCU0
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 SDHC1	SDHC1 interrupt status 0 No interrupt from SDHC1 1 Interrupt generated from SDHC1
27 SDHC0	SDHC0 interrupt status 0 No interrupt from SDHC0 1 Interrupt generated from SDHC0
26 DRAMC	DRAMC interrupt status 0 No interrupt from DRAMC 1 Interrupt generated from DRAMC
25 QSPI1	QSPI1interrupt status 0 No interrupt from QSPI1 1 Interrupt generated from QSPI1
24 QSPI0	QSPI0 interrupt status 0 No interrupt from QSPI0 1 Interrupt generated from QSPI0
23 RESERVED	This field is reserved. Reserved
22 Reserved	This field is reserved.
21 WDOGM4	WDOGM4 interrupt status 0 No interrupt from WDOGM4 1 Interrupt generated from WDOGM4
20 WDOGA5	WDOGA5 interrupt status 0 No interrupt from ADOGA5 1 Interrupt generated from WDOGA5
19–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

15.3.9 Interrupt Status Register 2 (GPC_ISR2)

This register indicates the source of interrupt causing system wake up during STOP mode.

Address: 4006_C000h base + 58h offset = 4006_C058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UART2	UART1	UART0	MLB	Flex		0	DAC1	DAC0	ADC1	ADC0	0	ANADIGA	ANADIGB	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		FTM				0	LPTMR	PIT	0		LCD	RLE	Reserved	0	VIU
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR2 field descriptions

Field	Description
31 UART2	UART2 interrupt status 0 No interrupt from UART2 1 Interrupt generated from UART2
30 UART1	UART1 interrupt status 0 No interrupt from UART1 1 Interrupt generated from UART1
29 UART0	UART0 interrupt status 0 No interrupt from UART0 1 Interrupt generated from UART0
28 MLB	MLB interrupt status 0 No interrupt from MLB 1 Interrupt generated from MLB
27–26 Flex	Flex interrupt status 0 No interrupt from Flex 1 Interrupt generated from Flex
25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 DAC1	DAC1 interrupt status

Table continues on the next page...

GPC_ISR2 field descriptions (continued)

Field	Description
	0 No interrupt from DAC1 1 Interrupt generated from DAC1
23 DAC0	DAC0 interrupt status 0 No interrupt from DAC0 1 Interrupt generated from DAC0
22 ADC1	ADC1interrupt status 0 No interrupt from ADC1 1 Interrupt generated from ADC1
21 ADC0	ADC0 interrupt status 0 No interrupt from ADC0 1 Interrupt generated from ADC0
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 ANADIGA	ANADIGA interrupt status 0 No interrupt from ANADIGA 1 Interrupt generated from ANADIGA
18 ANADIGB	ANADIGB interrupt status 0 No interrupt from ANADIGB 1 Interrupt generated from ANADIGB
17–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–10 FTM	FTM interrupt status 0 No interrupt from FTM 1 Interrupt generated from FTM
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 LPTMR	LPTMR interrupt status 0 No interrupt from LPTMR 1 Interrupt generated from LPTMR
7 PIT	PIT interrupt status 0 No interrupt from PIT 1 Interrupt generated from PIT
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 LCD	LCD interrupt status 0 No interrupt from LCD 1 Interrupt generated from LCD
3 RLE	RLE interrupt status

Table continues on the next page...

GPC_ISR2 field descriptions (continued)

Field	Description
	0 No interrupt from RLE 1 Interrupt generated from RLE
2 Reserved	Reserved This field is reserved.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 VIU	VIU interrupt status 0 No interrupt from VIU 1 Interrupt generated from VIU

15.3.10 Interrupt Status Register 3 (GPC_ISR3)

This register indicates the source of interrupt causing system wake up during STOP mode.

Address: 4006_C000h base + 5Ch offset = 4006_C05Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CCM_B	CCM_A	0	WKPU0	CMU	ASRC	SPDIF	ESAI	UNIMPLEMENTED				NFC	ESW	1588T1	1588T0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ENET1	ENET0	0	USBC1	USBC0	I2C3	I2C2	I2C1	I2C0	DSPI3	DSPI2	DSPI1	DSPI0	UART5	UART4	UART3
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR3 field descriptions

Field	Description
31 CCM_B	CCM_B interrupt status 0 No interrupt from CCM_A 1 Interrupt generated from CCM_A
30 CCM_A	CCM_A interrupt status 0 No interrupt from CCM_A 1 Interrupt generated from CCM_A

Table continues on the next page...

GPC_ISR3 field descriptions (continued)

Field	Description
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 WKPU0	WKPU0 interrupt status 0 No interrupt from WKPU0 1 Interrupt generated from WKPU0
27 CMU	CMU interrupt status 0 No interrupt from CMU 1 Interrupt generated from CMU
26 ASRC	ASRC interrupt status 0 No interrupt from ASRC 1 Interrupt generated from ASRC
25 SPDIF	SPDIF interrupt status 0 No interrupt from SPDIF 1 Interrupt generated from SPDIF
24 ESAI	ESAI interrupt status 0 No interrupt from ESAI 1 Interrupt generated from ESAI
23–20 UNIMPLEMENTED	This field is unimplemented. Read or write on this field has no effect.
19 NFC	NFC interrupt status 0 No interrupt from NFC 1 Interrupt generated from NFC
18 ESW	ESW interrupt status 0 No interrupt from ESW 1 Interrupt generated from ESW
17 1588T1	1588T1 interrupt status 0 No interrupt from 1588T1 1 Interrupt generated from 1588T1
16 1588T0	1588T0 interrupt status 0 No interrupt from 1588T0 1 Interrupt generated from 1588T0
15 ENET1	ENET1 interrupt status 0 No interrupt from ENET1 1 Interrupt generated from ENET1
14 ENET0	ENET0 interrupt status 0 No interrupt from ENET0 1 Interrupt generated from ENET0

Table continues on the next page...

GPC_ISR3 field descriptions (continued)

Field	Description
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 USBC1	USBC1 interrupt status 0 No interrupt from USBC1 1 Interrupt generated from USBC1
11 USBC0	USBC0 interrupt status 0 No interrupt from USBC0 1 Interrupt generated from USBC0
10 I2C3	I2C3 interrupt status 0 No interrupt from I2C3 1 Interrupt generated from I2C3
9 I2C2	I2C2 interrupt status 0 No interrupt from I2C2 1 Interrupt generated from I2C2
8 I2C1	I2C1 interrupt status 0 No interrupt from I2C1 1 Interrupt generated from I2C1
7 I2C0	I2C0 interrupt status 0 No interrupt from I2C0 1 Interrupt generated from I2C0
6 DSPI3	DSPI3 interrupt status 0 No interrupt from DSPI3 1 Interrupt generated from DSPI3
5 DSPI2	DSPI2 interrupt status 0 No interrupt from DSPI2 1 Interrupt generated from DSPI2
4 DSPI1	DSPI1 interrupt status 0 No interrupt from DSPI1 1 Interrupt generated from DSPI1
3 DSPI0	DSPI0 interrupt status 0 No interrupt from DSPI0 1 Interrupt generated from DSPI0
2 UART5	UART5 interrupt status 0 No interrupt from UART5 1 Interrupt generated from UART5
1 UART4	UART4 interrupt status

Table continues on the next page...

GPC_ISR3 field descriptions (continued)

Field	Description
	0 No interrupt from UART4 1 Interrupt generated from UART4
0 UART3	UART3 interrupt status 0 No interrupt from UART3 1 Interrupt generated from UART3

15.3.11 Interrupt Status Register 4 (GPC_ISR4)

This register indicates the source of interrupt causing system wake up during STOP mode.

Address: 4006_C000h base + 60h offset = 4006_C060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0	0				RESERVED			0	EWM	PDB	SRC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR4 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 GPIO4	GPIO4 interrupt status 0 No interrupt from GPIO4 1 Interrupt generated from GPIO4
14 GPIO3	GPIO3 interrupt status 0 No interrupt from GPIO3 1 Interrupt generated from GPIO3
13 GPIO2	GPIO2 interrupt status 0 No interrupt from GPIO2 1 Interrupt generated from GPIO2

Table continues on the next page...

GPC_ISR4 field descriptions (continued)

Field	Description
12 GPIO1	GPIO1interrupt status 0 No interrupt from GPIO1 1 Interrupt generated from GPIO1
11 GPIO0	GPIO0 interrupt status 0 No interrupt from GPIO0 1 Interrupt generated from GPIO)
10–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 RESERVED	This field is reserved.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 EWM	EWM interrupt status 0 No interrupt from EWM 1 Interrupt generated from EWM
1 PDB	PDB interrupt status 0 No interrupt from PDB 1 Interrupt generated from PDB
0 SRC	SRC interrupt status 0 No interrupt from SRC 1 Interrupt generated from SRC

15.4 Functional Description

15.4.1 Power Shutdown Controller

The Power Shutdown Controller takes control once all the clocks are successfully gated and GPC is configured for LPSTOPn modes. It initiates the power down request to Power Gating Controller (PGC) enabling PGC to start the power down sequence. It also monitors wake up request and accordingly generates power up request to PGC after HPREG stablizes. The system can wake-up from the following sources

- External reset
- System wake-up from WKPU
- Level Trigger Interrupt (only for STOP mode)

NOTE

During wake-up by interrupt, software should mask the corresponding interrupt in GPC_IMR register which are not the source of wake-up during STOP mode.

15.4.2 Power Gating Controller

The Power Gating Controller (PGC) is a power management component that controls the power-down and power-up sequencing of individual subsystems. For subsystems to be completely powered down in low power modes, a specific sequence of power control signals must be followed. The sequence timing is programmable using the PGC control registers.

Figure 15-12 shows the PGC as part of the overall power management scheme in this device.

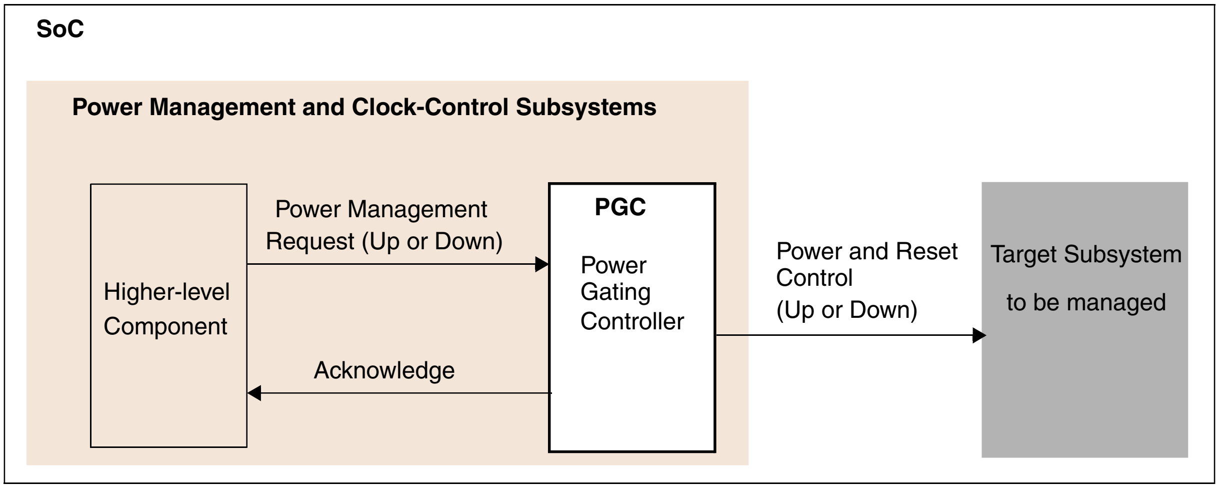


Figure 15-12. PGC Top-Level Block Diagram

15.4.2.1 Functional Mode

The following sections describe the power-control sequencing steps for various low power modes .

15.4.2.1.1 LPSTOP Entry Sequence (PG Entry Sequence)

These steps outline the power-down sequence:

1. Software sets the GPC_PGCR[PG_PD1] bit, see Power Gating Control Register (GPC_PGCR).
2. Software also configures the GPC_LPMR[CLPCR] to STOP mode.

3. CCM stops the clocks to the target subsystem on receiving ARM_DSM_REQUEST after executing WFI instruction.
4. On receiving the stop clock indication from CCM, GPC put the memories into deep sleep if GPC_PGCR[DS_LPSTOP] is set to '1'.
5. GPC asserts *isolation* to isolate the outputs of power-gated domain.
6. GPC triggers SRC to assert reset to power gated domain(PD1) .
7. VREG_DIG then power gated the power domains after 20 μs.
8. VREG_DIG also powered down HPREG and LPREG after 50 μs.
9. FIRC is disabled if CCM_CCR[FIRC_EN] is configured to '0' after VREG_DIG completely power down the regulators.

Figure 15-13 shows the power-down sequence.

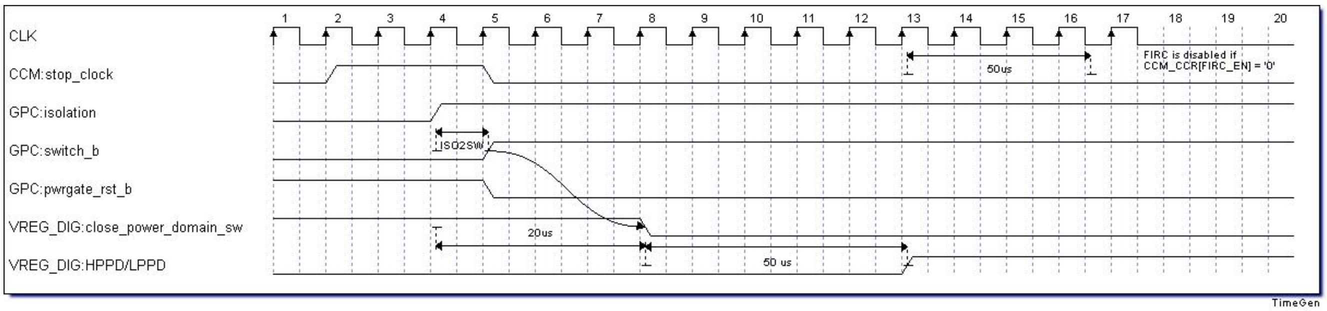


Figure 15-13. Power-Down Sequence Timing Diagram

15.4.2.1.2 LPStop Mode Exit Sequence

The LPStop mode exit sequence is not interruptible and initiated only on receiving system wake up, or external reset.

These steps outline the LPStop exit sequence:

1. On exit, GPC first enables the FIRC and removes the deep sleep synchronously
2. VREG_DIG enables the HPREG and LPREG on receiving wake-up indication from GPC
3. GPC, then enables all the power domain switches after HP regulator stabilizes.
4. GPC indicates SRC to remove the reset from power gated domain once all the power domain switches are closed.
5. GPC negates the isolation after two cycles.
6. CCM then starts enabling the clocks once the isolation is removed.

Figure 15-14 shows the power-up sequence.

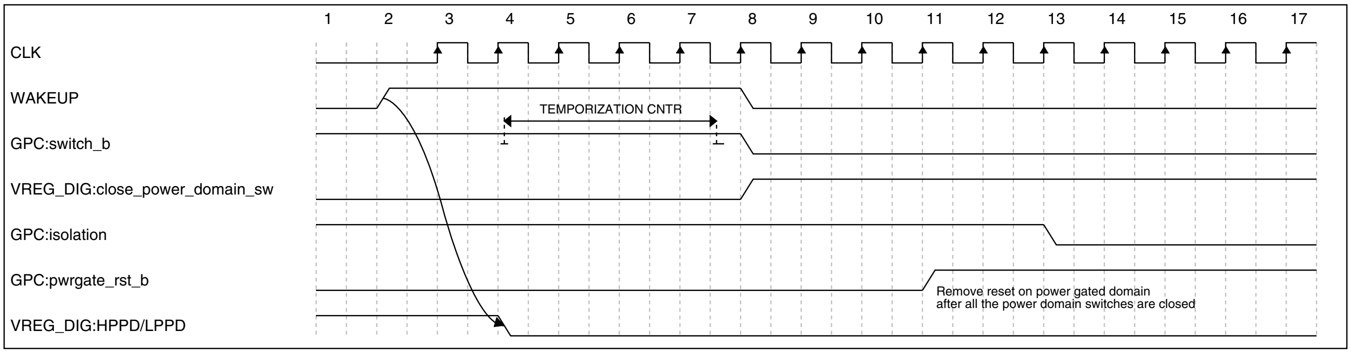


Figure 15-14. Power-Up Sequence Timing Diagram

15.4.2.1.3 Stop Mode Entry Sequence

The following steps outline the normal stop mode entry sequence:

1. Software ensures GPC_PGCR[PG_PD1] bit and GPC_PGCR[WB_STOP] bit are not set; see Power Gating Control Register (GPC_PGCR)
2. Software also configures the GPC_LPMR[CLPCR] to STOP mode.
3. The clock-control subsystem stops the clocks to the target subsystem on receiving ARM_DSM_REQUEST after executing WFI instruction.
4. On receiving stop clock indicating from the CCM, HPREG is power-down and put the memories into deep-sleep based on GPC_PGCR[HP_OFF] and GPC_PGCR[DS_STOP] configurable bit.
5. FIRC is disabled, if CCM_CCR[FIRC_EN] is configured to '0'.

15.4.2.1.4 Stop Mode Exit Sequence

The following steps outline the normal stop mode exit sequence on receiving system wake up, external reset or interrupt which are configured as source of wakeup:

1. The stop-mode exit is initiated by enabling the FIRC oscillator
2. HPREG is enabled synchronously and deep-sleep is removed
3. After the expiry of 10us (configurable through fuse), wakeup is generated to CCM to enable the clocks

15.4.2.1.5 Well Bias Stop mode entry sequence

These steps outline the well bias stop mode sequence:

1. Software ensures GPC_PGCR[PG_PD1] bit is not set and GPC_PGCR[WB_STOP] bit is set; see Power Gating Control Register (GPC_PGCR)
2. Software also configures the GPC_LPMR[CLPCR] to STOP mode.
3. The clock-control subsystem stops the clocks to the target subsystem
4. On receiving stop clock indicating from the CCM, well bias cell are isolated and put the memories in deep sleep based on GPC_PGCR[DS_STOP].

5. PMOS switch is then disabled after 2 clocks
6. After 5us, Well Bias Regulator is enabled.
7. HPREG is powered down after 5 us once the WB regulator is enabled. Also disabled FIRC if CCM_CCR[FIRC_EN] is configured to '0'.

Refer to Voltage Regulators for details.

15.4.2.1.6 Well Bias Stop mode exit sequence

The well bias stop mode exit is initiated on receiving system wake up, or external reset. It also exits from well bias stop mode on interrupts which are configured as source of wake up from stop mode. The following outline the well bias stop mode exit sequence.

1. On well bias stop mode exit indication,GPC enables FIRC oscillator if it was configured to be off.
2. HPREG is then enabled synchronously and deep-sleep is also removed
3. Wakeup is generated to CCM after 300 ns
4. After 5us, Well Bias regulator is disabled
5. PMOS switch is enabled after the expiry of another 5us and then isolation is removed after 2 clocks
6. CCM sampled the wakeup signal once isolation is removed and enabled the clocks henceforth

15.4.2.1.7 Inhibit Stop and FIRC Control

The inhibit stop signal from the platfom prevents the entry into STOP mode. This is possible by overwriting the LPM mode provided to the CCM. During inhibit stop, the LPM mode exposed to CCM is RUN mode irrespective of other mode configured in GPC_LPMR register. The inhibit stop signal is discarded if the GPC is configured for LPSTOP mode (where GPC_LPMR[CLPCR] = 2’b10 and GPC_PGCR[PG_PD1] = 1’b1. See Low Power Mode Register (GPC_LPMR) and Power Gating Control Register (GPC_PGCR)

The FIRC oscillator can be disabled during LPRUN and STOP mode to minimize power. During LPRUN GPC_LPMR[CLPCR = 2'b00], enabling and disabling of FIRC is software controllable through CCM_CCR[FIRC_EN] bit. When GPC_LPMR[CLPCR] is configured to WAIT/STOP mode, FIRC is enabled by default and it gets disabled only after the completion of Low-power mode entry by hardware depending on CCM_CCR[FIRC_EN] configurable bit in CCM. On exit, FIRC is enabled and remained ON until GPC_LPMR[CLPCR] is configured to RUN mode and CCM_CCR[FIRC_EN] is configured to '0'.

Chapter 16
Voltage Regulators

16.1 Overview

The power blocks are used for providing 1.2 V digital supply to the Core logic of this device. The main/input supply voltage is 3.3 V. Digital/regulated output supply is 1.20 V - 1.26 V, for High Power/Run mode. Maximum load current supported is 1200 mA in high power mode. There are two other modes, namely "low power" and "ultra low power" with lower current driving capabilities and lower power consumption to support the device low power modes.

Power management for this device requires the following blocks described below:

- High power regulator HPREG
- Low Power Regulator LPREG
- Ultra Low Power Regulator ULPREG
- 1.1 V Regulator LDO_IP1
- 2.5 V Regulator LDO_2P5
- 3.0 V Regulator USB_LDO
- SNVS Regulator
- Voltage reference for HPREG
- Voltage reference for LPREG, ULPREG, and Low Voltage Detectors (LVD)
- Low Voltage detector for 3.3 V supply
- Separate Low voltage detector for HPREG, LPREG, and ULPREG output voltages
- Separate High voltage detector (HVD) for HPREG
- Power on Reset (POR)

- Power-up sequencing
- N-well Bias circuit, which contains Well Bias Regulator (WBREG), to reduce leakage power in Low Power modes.

The voltage regulator digital interface provides the temporization delay of 300 μs at initial power-up and and during exit from LPSTOP mode. This, in conjunction with POR1 (which means ULPREG is powered-up), indicates the SRC to release the reset to the device and enter to the next phase, i.e, phase1. It provides trimming controls for regulators and LVDs. It also manages transition from run to low power mode and vice-versa. It generates interrupt on High Voltage Detection during RUN and STOP mode.

Various feature of the VREG digital interface are:

- Temporization counter of 300 μs to get HPREG and HPBGAP up and stable during:
  - initial power up
  - LPSTOP mode exit
- Transition between different modes:
  - RUN to STOP
  - RUN to LPSTOP mode
  - STOP to RUN mode
  - LPSTOP to RUN mode
- Source of Power-On-Reset to Sytem Reset Controller.
- Interrupt on HVD (High Voltage Detection) during RUN and STOP mode
- Masking of HVD and HPLVD (High Power Low Voltage Detection) during LPSTOP mode
- Well Bias circuit control during STOP mode

16.1.1 Signal Description

The following signals connect off-chip. VDDREG, VSS, VDD are the Power supplies.

Table 16-1. Signal Description

Signal name	Direction	Description
VDDREG	In/Out	3.3 V supply for regulation

Table continues on the next page...

Table 16-1. Signal Description (continued)

Signal name	Direction	Description
VSS	In/Out	Ground for Supplies
BCTRL	Out	Base control Pin for the external Ballast transistor.
VDD	In/Out	1.2V regulated supply from Ballast transistor.

16.1.2 Digital Interface Block Diagram

The following figure shows the block diagram for Voltage Regulator Digital Interface.

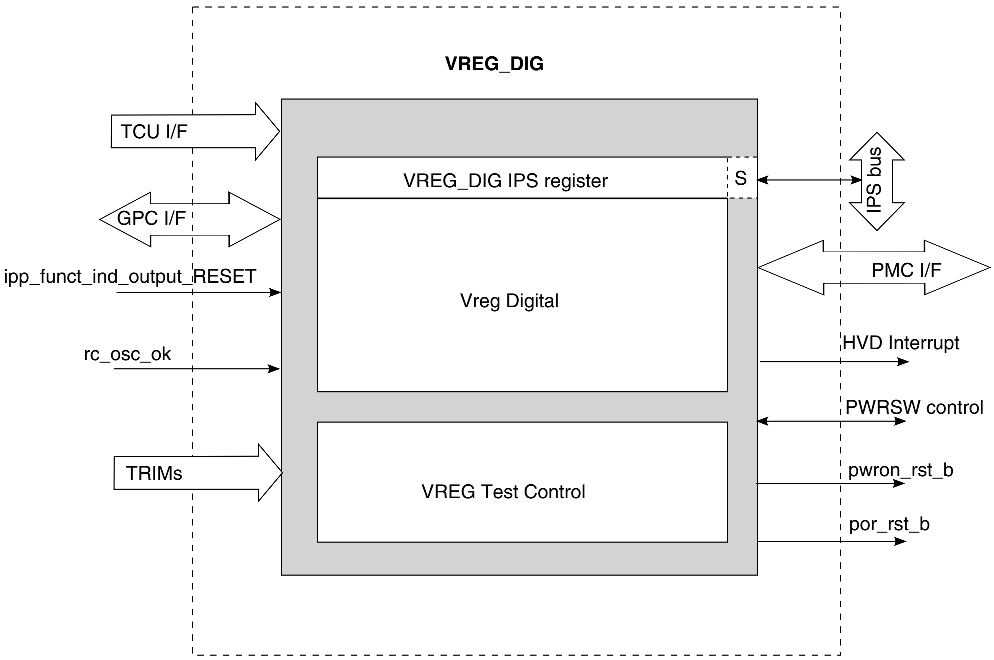


Figure 16-1. VREG_DIG Block Diagram

16.2 Memory Map and Registers

VREG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_D000	Control Register (VREG_CTRL)	32	R/W	0000_0003h	16.2.1/780
4006_D004	Status register (VREG_STAT)	32	R/W	0000_0000h	16.2.2/781

16.2.1 Control Register (VREG_CTRL)

The lower bit (Mask) of this register controls the masking of the HVD interrupt. The reset value of this bit is such that it unmasks the HVD interrupt.

The PORPU bit is used to control PORPU which is an input 1.2V level active high signal of the voltage regulator and is used to disable the POR-LVDMOKHV loop. The signal controls the zero/non-zero current consumption mode of the POR. The signal can control the POR only after LVDMOKHV is high. To save current consumption in low power mode (STOP/LPSTOP), this circuitry can be disabled by driving 0 on PORPU. This will disable the generation of POROUT. Whenever the High Power regulator is switched-off, POROUT circuitry can also be disabled by programming this register bit. This register can be programmed anytime by the software but the affect would only happen when the HPPD to voltage regulator is 1 (that is, HPREG is off). During RUN mode this circuitry is always ON as it does not provide considerable current consumption.The lower bit (Mask) of this register controls the masking of the HVD interrupt. The reset value of this bit is such that it unmasks the HVD interrupt.

Address: 4006_D000h base + 0h offset = 4006_D000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															PORPU
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												HVD_PD	POR1P2_PD	0	HVDMASK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

VREG_CTRL field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 PORPU	This bit is used to control the i/p signal PORPU of voltage regulator during STOP/LPSTOP mode. 1 : POR circuitry is enabled 0 : POR circuitry is disabled.

Table continues on the next page...

VREG_CTRL field descriptions (continued)

Field	Description
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 HVD_PD	HVD Power Down.
2 POR1P2_PD	P0R1P2 Power Down.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 HVDMASK	Mask HVD interrupt. 1 : Enable mask 0 : Disable mask

16.2.2 Status register (VREG_STAT)

The lower bit (HVDSTAT) provides the status of the HVD interrupt. It is set whenever high voltage is detected. It is usually masked during Low Power Stop mode irrespective of HVDMASK in the VREG_CTRL register.

Address: 4006_D000h base + 4h offset = 4006_D004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															HVDSTAT
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VREG_STAT field descriptions

Field	Description
31–1 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
0 HVDSTAT	This bit status of HVD. It is read only and write 1 to clear. 1 : HVD interrupt is set 0 : HVD interrupt is cleared

16.3 Functional Description

The voltage regulator consists of three internal regulators - HPREG, LPREG and ULPREG. These are designed keeping in mind that when in LPSTOP mode, the HPREG and LPREG are turned off; while in STOP mode, HPREG is turned off. The always-ON design works on ULPREG. In RUN mode, the main current is supplied from HPREG (trimmed at 1.20V) and the LPREG and ULPREG (trimmed at 1.15V) are unloaded.

16.3.1 High Power or Main Regulator (HPREG)

This section describes High Power Regulator (HPREG), which is used to convert 3.3 V input supply (VDDREG) to 1.23 V digital supply. HPREG is split into two parts:

- Regulator Controller
- The NPN bipolar transistor, which is placed external to the chip

The base of the bipolar transistor is driven by a control signal generated by the controller. The nominal target output is 1.23 V typical. The actual output will be in the range of 1.20 V - 1.26 V, post trimming. The stabilization for HPREG is achieved using an external capacitance. The minimum recommended value is 4.7 µF. HPREG gets its reference voltage from a Bandgap reference circuit, which is internal to the module (LPBGAP, with a voltage of 1.2V post trim). This reference voltage is required to be trimmed to reduce the spread of the output supply.

16.3.2 Low Power Regulator (LPREG)

LPREG generates power for the device in STOP mode and has a drive strength of 200mA. The output of this regulator is tied to HPREG and in normal mode it does not supply any load current. The regulator is driven by VDDREG supply and provides the output supply of 1.22V in STOP mode, when HPREG is turned off. It has programmable output voltage in steps of 25mV.

Post trim output is in the range of 1.18 V to 1.24 V.

16.3.3 Ultra Low Power Regulator (ULPREG)

ULPREG generates power for the Low Power Stop modes when both HPREG & LPREG are turned off. Output of this regulator is tied those of HPREG & LPREG, but it does not supply current in normal or STOP mode. It has very low power consumption and has low load and line regulation requirements. It has output voltage programmability, but no offset programmability. Maximum drive current is 20mA.

Internal ballasts are also present in ULPREG.

16.3.4 LVDs and POR

Two types of LVDs are available:

- LVD_MAIN for the 3.3V input supply with upper and lower thresholds at 2.770 V and 2.748 level.
- Three separate LVD_DIG for the 1.2V common output voltage of HPREG, LPREG and ULPREG

The LVD_MAIN is for VPLUSIO supply and provides ULVDMOKHV and ULVDMOKLV as active high signals at 3.3V and 1.2V supply levels, respectively. One LVD_DIG is at HPREG output, one at LPREG and one at output of ULPREG. MLVDDOKHV/LV are active high signals for LVD at HPREG,and ULVDMLPOKHV/LV for LVD at LPREG . LVD_DIG placed in the standby domain at ULPREG output senses the standby ULPVDD supply level. It provides ULVDDOKHV/LV as active high signals. The reference voltage used for all LVDs is from the trimmed LPBGAP. All LVD_DIG can be programmed for different thresholds. Power-down pins are provided for all LVDs.

When LVDs are powered down, their outputs are pulled high. When high power and low power regulators are OFF, the LVD_DIG placed in the main domain should also be in power-down state.

16.3.5 Power Up Sequencing

During supply ramp-up, POR1P2 is used to mask all 1.2V control signals and initialize the LVDOKs to '0' output level. At start-up, all regulators and LVDs are turned ON after 3.3V supply POR3P3 is released. The 1.2V control bit masking is released once:

- POR1P2 is released for 1.2V supply which is after 1.2V supply is stable
- ULVDDOKHV, MLVDDOKHV, and ULVDMOKHV are '1'

During start-up or LPSTOP exit mode, load should be applied only after waiting for a time equal to the start-up time of HPREG. During the normal operation, all of the three regulators HPREG, LPREG and ULPREG will be simultaneously on, but the load current will be supplied by HPREG only. Similarly, when the device enters STOP mode, HPREG will be off, but LPREG and ULPREG will be simultaneously on, and the load current will be supplied by LPREG. During LPSTOP mode, only ULPREG will be on and LPREG and HPREG will be turned off.

The following sections provides the functional information of VREG_DIG and how the voltage regulator should be controlled during various low-power modes.

16.3.6 STOP Mode

When the device is operating in RUN mode, HPREG and LPREG both are ON, but current is supplied by the HPREG. In the Stop mode, device is powered up but there is not much activity (clock) happening inside the device. Therefore, in the Stop mode, LPREG is sufficient to provide the required device current and hence, HPREG is switched off. On system wakeup, it enables the HPREG and switch to RUN mode.

16.3.7 Low Power Stop Mode (LPSTOP)

In LPSTOP mode, a small part of the device is ON and the remaining part is not powered. In this mode, ULPREG supplies the core current (leakage) while HPREG and LPREG is OFF. VREG_DIG also opens all the power switches (PD1, 16K and 48K) based on the configuration provided in GPC power gate control register before powering

off the HPREG and LPREG. On system wakeup, it enables the transition to RUN mode by enabling the HPREG and LPREG. All the power switches are closed after HPREG stabilizes.

16.3.8 Well Bias STOP Mode

It enables the system to enter into lowest power STOP mode. In this mode, the device current is provided by WBREG regulator while HPREG is switched off. On system wakeup, it enables the HPREG and disables the WBREG.

Chapter 17
Reset

17.1 Introduction

This chapter describes the Reset architecture for this device.

17.2 Reset Sources

The possible sources of reset on this device are as follows:

- POR (Power On Reset)
- External hardware reset via the /RESET pin (ext_reset_b pin) - Active Low
- Software JTAG
- Watchdog Reset
- Security Reset (CSU_RESET_B)
- Clock Monitor Reset (CMU). The CMU is used to monitor the main 24MHz crystal oscillator.
- PMU Low Voltage Detection. LVDs on the 3.3V Main supply and the 1.2V core supplies will trigger a reset event. HVDs on the supplies do not trigger a reset but will optionally trigger an interrupt.
- CPU Exception Error Check (illegal opcode or other CPU exception error)

17.3 Reset Functions

The following table shows the action on the SoC of each of the possible reset sources. Although the LDOs do not generate resets they are listed in the table. Only the main PMU can generate a reset based on its LVDs.

The various 'targets' for reset are grouped as follows:

A -> All device Configurations (Trimbits, fuse shadow bits, memory repair registers), Boot configuration & reset vectors, CCM, OCOTP, ANADIG.
B -> SJC, TAP registers
C -> Security Logic
D -> SRTC
E -> DDR Logic
F ->All other logic on SoC

Table 17-1. Reset Functionality

	A	B	D	E	F	
Reset Source	**Configs**	**SJC/TAP**	**RTC**	**DDR Logic**	**All other**	**Comments**
POR	Reset	Reset	Reset	Reset	Reset	Reset all with POR
/RESET	Reset	Reset	No	Reset	Reset	RTC runs through Reset
Software JTAG	Reset	No	No	Reset	Reset	
WDOG	No	No	No	Reset	Reset	
CMU	Reset	Reset	Reset	Reset	Reset	CMU monitors loss of 24MHz XTAL
LVD (PMU)	Reset	Reset	Reset	Reset	Reset	
HVD (PMU)	No	No	No	No	No	Configurable interrupt, no reset
LDO-OK (3.3/1.1-analog)	No	No	No	No	No	Configurable interrupt, no reset
LDO-OK (3.3/2.5-analog)	No	No	No	No	No	Configurable interrupt, no reset
LDO-OK (3.3/1.1-coin)	No	No	No	No	No	No interrupt relevant as only VBat domain alive
LDO-OK (5.0/3.3-USB)	No	No	No	No	No	Configurable interrupt, no reset
CPU Exception	No	No	No	Reset	Reset	

17.4 Clock Monitor

The Clock Monitor Unit (CMU) is used to monitor the main 24MHz crystal oscillator. Upon failure of the crystal the CMU should reset the system.

Chapter 18
System Reset Controller (SRC)

18.1 Introduction

The SRC - System Reset Controller- generates the resets for the whole device. The functional reset sources are programmable as either reset or interrupt. The block also generates interrupts for various device events.

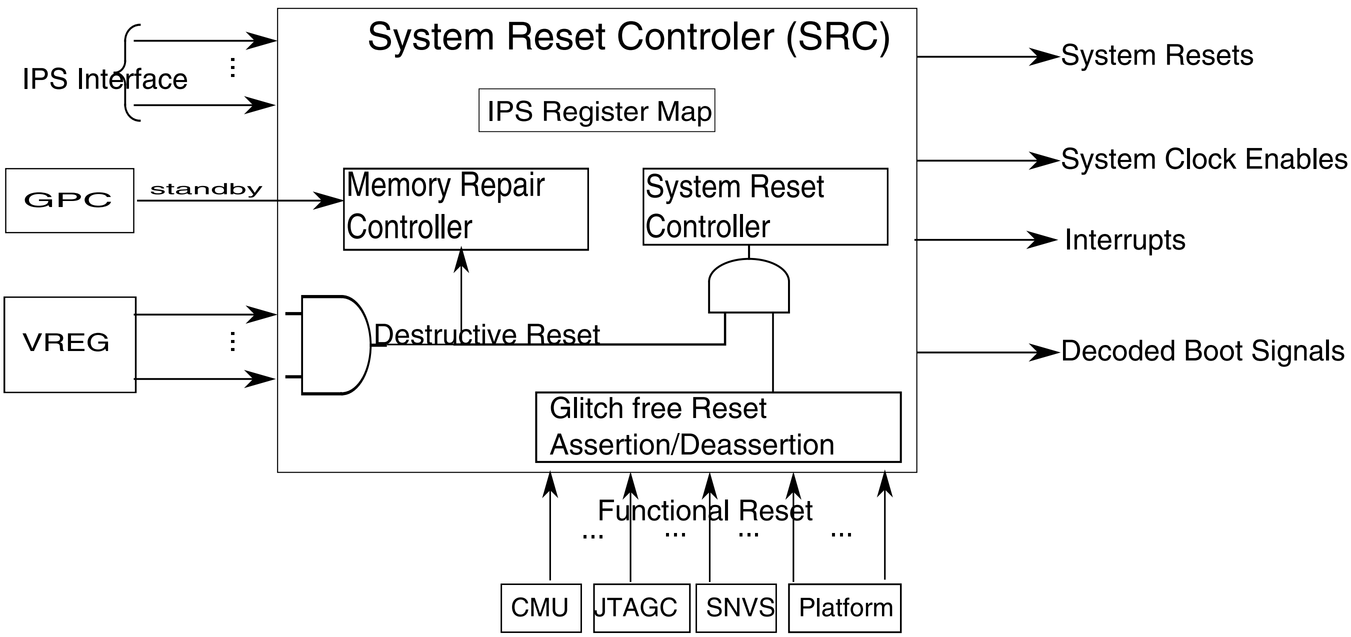


Figure 18-1. Functional reset sequence

18.2 SRC Overview

The system reset controller (SRC) generates all reset signals for the device. The SRC controls the bringup of the device from all of the destructive and functional resets. The SRC handles three types of resets

- Destructive reset: These resets are caused by LVD(s) and poweron reset. It causes the whole device to reset.
- Functional reset: These resets are caused by different events. They cause most of the system to reset except the modules resettable only by destructive reset like SRC, DAP, WDOG, etc.
- Standby reset: This reset is triggered by programming the GPC. The power domain 1 of the system is reset.

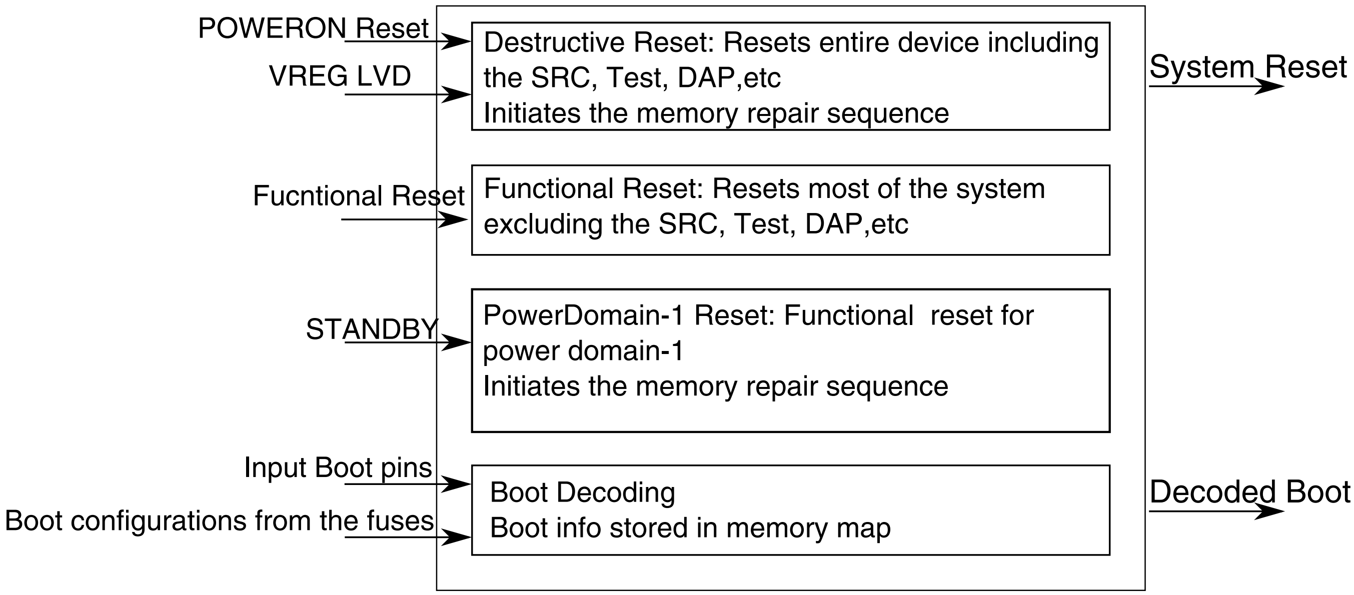


Figure 18-2. SRC High Level Diagram

18.2.1 Features

The SRC includes the following features.

- Receives and handles the resets from all the reset sources
- Resets the appropriate domains based upon the reset sources and the nature of the reset
- Latches the BMOD pins and common configuration signals from the internal fuses
- Provides user flexibilty to choose the events as functional reset or interrupt
- Latches all reset/interrupt events in the SRC_SRSR

18.3 Memory Map and Register Definition

18.3.1 Register Descriptions

This section consists of register descriptions in address order. All registers are reset on POR sequence.

SRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_E000	SRC Control Register (SRC_SCR)	32	R/W	0000_050Ah	18.3.2/792
4006_E004	SRC Boot Mode Register 1 (SRC_SBMR1)	32	R/W	0000_0000h	18.3.3/793
4006_E008	SRC Status Register (SRC_SRSR)	32	R/W	00FE_FF65h	18.3.4/793
4006_E00C	SRC_SECR	32	R/W	0000_0000h	18.3.5/797
4006_E014	SRC Reset Interrupt Configuration Register (SRC_SICR)	32	R/W	FF00_0002h	18.3.6/798
4006_E018	SRC Interrupt Masking Register (SRC_SIMR)	32	R/W	00FF_FFFDh	18.3.7/800
4006_E01C	SRC Boot Mode Register 2 (SRC_SBMR2)	32	R/W	0000_0000h	18.3.8/802
4006_E020	General Purpose Register (SRC_GPR0)	32	R/W	0000_0000h	18.3.9/803
4006_E024	General Purpose Register (SRC_GPR1)	32	R/W	0000_0000h	18.3.9/803
4006_E028	General Purpose Register (SRC_GPR2)	32	R/W	0000_0000h	18.3.9/803
4006_E02C	General Purpose Register (SRC_GPR3)	32	R/W	0000_0000h	18.3.9/803
4006_E030	General Purpose Register (SRC_GPR4)	32	R/W	0000_0000h	18.3.9/803
4006_E04C	MISC0 (SRC_MISC0)	32	R (reads 0)	0000_0000h	18.3.10/ 803
4006_E050	MISC1 (SRC_MISC1)	32	R/W	0000_0000h	18.3.11/ 805
4006_E054	MISC2 (SRC_MISC2)	32	R/W	0000_0000h	18.3.12/ 807
4006_E058	MISC3 (SRC_MISC3)	32	R/W	0000_0000h	18.3.13/ 808

18.3.2 SRC Control Register (SRC_SCR)

The following figure presents the Reset control register (SCR), which contains bits that control operation of the reset controller and the table provides its field descriptions.

Address: 4006_E000h base + 0h offset = 4006_E000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			0	0	CM4_WDGRST_MASK				0			CA5_WDGRST_MASK			
W				SW_RST												
Reset	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0

SRC_SCR field descriptions

Field	Description
31–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 SW_RST	Software reset. Initiates Functional reset sequence 0 Software reset not requested 1 Software reset requested Self clearing, Always read 0
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–7 CM4_WDGRST_ MASK	Mask wdog_rst_b of CM4 core. If these 4 bits are coded from A to 5 ,then wdog_rst_b input to SRC will be masked and wdog_rst_b will not create reset to the device. Any other code will be coded to 1010 that is wdog_rst_b is not masked. 0101 wdog_rst_b is masked 1010 wdog_rst_b is not masked (default) Write of anyvalue other than 01010 will set the field to 1010
6–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 CA5_WDGRST_ MASK	Mask wdog_rst_b of CA5 source. If these 4 bits are coded from A to 5 then, wdog_rst_b input to SRC will be masked and wdog_rst_b will not create reset to the IC. 0101 wdog_rst_b is masked 1010 wdog_rst_b is not masked (default) any other code will be coded to 1010 i.e. wdog_rst_b is not masked **NOTE:** During the time the WDOG event is masked using SRC logic, it is likely that WDOG Reset Status Register (WRSR) bit 1 (which indicates WDOG timeout event) will get asserted. SW / OS developer must prepare for this case. Re-enable WDOG is possible, by un-mask it in SRC, though it must be preceded by servicing the WDOG. However, for the case that the event has been asserted, the status bit (WRSR bit-1) will remain asserted, regardless of servicing the WDOG module. (HW reset is the only mean to cause de-assertion of that bit).

Table continues on the next page...

SRC_SCR field descriptions (continued)

Field	Description
	0101 wdog_rst_b is masked 1010 wdog_rst_b is not masked (default)

18.3.3 SRC Boot Mode Register 1 (SRC_SBMR1)

The following presents the Boot Mode register (SBMR1), which contains bits that reflect the status of Boot Mode Pins of the chip. The default values for those bits depend on the values of pins/fuses during reset sequence.The boot configuration signals are sampled once fuses are read. The boot configuration can either be sampled from OCOTP or GPIO based on the boot mode selected. The table provides its field descriptions.

NOTE

When sampled from GPIO, these bits are only updated on a Power On Reset (POR).

Address: 4006_E000h base + 4h offset = 4006_E004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BOOT_CFG4								BOOT_CFG3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BOOT_CFG2								BOOT_CFG1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_SBMR1 field descriptions

Field	Description
31–24 BOOT_CFG4	Please refer to OCOTP_CFG4 .
23–16 BOOT_CFG3	Please refer to OCOTP_CFG3 .
15–8 BOOT_CFG2	Please refer to OCOTP_CFG2 .
7–0 BOOT_CFG1	Please refer to OCOTP_CFG1 .

18.3.4 SRC Status Register (SRC_SRSR)

The SRSR is a status register that records the source of the reset events for the chip. The SRSR register bit is set when the event occurs.

NOTE

On a Power on Reset, the register will read 0x00FEFF65. The recommendation is to clear all reset sources after a POR event to read the proper reset status for future reset events.

NOTE

When using a debugger, the RESETB bit will also be set due to the debugger asserting reset.

NOTE

The following bits are expected to be valid after a power on reset.

- SRSR[POR_RST]
- SRSR[CA5_WDG_RST]
- SRSR[WDOG_RST]
- SRSR[RESET_B] - due to debugger
- SRSR[POR1P2]
- SRSR[HP_LVD]
- SRSR[ULP_LVD]
- SRSR[LVD_P3P]
- SRSR[LP_LVD]
- SRSR[MDM_SYS_RST]

At bootup the SW can mask-read all the other SRSR bits i.e. ignore other bits. All bits should be cleared by writing '1' to them after bootup for proper status setting.

Address: 4006_E000h base + 8h offset = 4006_E008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CM4_LCK	SOSC_OK	0	CMU_FLL_FHH	CMU_OLR	REG_3P0_OK	REG_2P5_OK	REG_1P1_OK	0				0	SRC_SW_RST	0	MDM_SYS_RST
W	w1c	w1c		w1c	w1c	w1c	w1c	w1c						w1c		w1c
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			LP_LVD	LVD_P3P	ULP_LVD	HP_LVD	POR1P2	RESETB	0	JTAG_RST	WDOG_RST		0	CA5_WDG_RST	POR_RST
W				w1c	w1c	w1c	w1c	w1c	w1c		w1c	w1c			w1c	w1c
Reset	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0	1

SRC_SRSR field descriptions

Field	Description
31 CM4_LCK	Indicates when CM4 is in lockup. 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset .
30 SOSC_OK	Indicates no clock is detected on SOSC 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 CMU_FLL_FHH	Indicates CMU event when BUS frequency clock is out of range 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset .
27 CMU_OLR	Indicates CMU event of FOSC frequency less than 40 Mhz 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
26 REG_3P0_OK	Indicates Anadig regulator 3.0V unstable event 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
25 REG_2P5_OK	Indicates Anadig regulator 2.5V unstable event 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
24 REG_1P1_OK	Indicates Anadig regulator 1.1V unstable event 1 The SRSR register bit is set when the event occurs. 0 The SRSR register bit is reset.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SRC_SRSR field descriptions (continued)

Field	Description
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 SRC_SW_RST	Indicates SRC_SCR[SW_RST] is set 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 MDM_SYS_RST	Indicates the MDM-AP[SYSRESETREQ] is set 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 LP_LVD	Indicates LP regulator's LVD event 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
11 LVD_P3P	Indicates 3.3V main supply is unstable 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
10 ULP_LVD	Indicates ULP regulator's LVD event 1 The SRSR register bit is set when the event occurs 0 reset.set when the event occurs
9 HP_LVD	Indicates HP regulator's LVD event 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
8 POR1P2	Indicates 1.2V supply goes below 0.7V event 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
7 RESETB	Indicates external reset event 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 JTAG_RST	Indicates HIGH-Z JTAG event 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
4–3 WDOG_RST	Indicates watchdog Time-out event. WDOG_RST_F[1] - CM4 core WDOG_RST_F[0] - CA5 core 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.

Table continues on the next page...

SRC_SRSR field descriptions (continued)

Field	Description
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 CA5_WDG_RST	Indicates event from internal CA5 watchdog timer 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.
0 POR_RST	Indicates event POR1- power on reset. 1 The SRSR register bit is set when the event occurs 0 The SRSR register bit is reset.

18.3.5 SRC_SECR

Address: 4006_E000h base + Ch offset = 4006_E00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											WRLOCK	0		SPIDEN	SPNIDEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_SECR field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 WRLOCK	Once writeable. Once written the SPIDEN and SPNIDEN can't be written
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 SPIDEN	Secure invasive debug enable. Can be set by user running device in secure mode. Refer to Additional Authentication Interface
0 SPNIDEN	Secure non-invasive debug enable. Can be set by user running device in secure mode.

18.3.6 SRC Reset Interrupt Configuration Register (SRC_SICR)

The SICR is a reset event configuration register. The event mentioned in the bitfield is configured as interrupt, when field is '1' and as reset when the bitfield value is '0'.

Address: 4006_E000h base + 14h offset = 4006_E014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CM4_LCK	0	FOSC_OK_CFG	CMU_FLL_FHH	CMU_OLR	1	REG_2P5	REG_1P1	RESERVED				0	0	0	0
W																
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									RESERVED	JTAG_RST	WDOG_RST		RESERVED	CA5_WDG_RST	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

SRC_SICR field descriptions

Field	Description
31 CM4_LCK	Configures when CM4 is in lockup 1 The bitfield is configured as interrupt 0 The bitfield is reset.
30 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
29 FOSC_OK_CFG	Configure ANADIG_ANA_MISC0[OSC_XTALOK_EN] register to 1 to support no_fosc clock as interrupt/reset event.
28 CMU_FLL_FHH	Configures CMU event when BUS frequency clock is out of range 1 The bitfield is configured as interrupt 0 The bitfield is reset.
27 CMU_OLR	Configures CMU event of FOSC frequency less than 40 Mhz 1 The bitfield is configured as interrupt 0 The bitfield is reset.
26 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 1.
25 REG_2P5	Configures Anadig regulator 2.5V unstable event

Table continues on the next page...

SRC_SICR field descriptions (continued)

Field	Description
	1 The bitfield is configured as interrupt 0 The bitfield is reset.
24 REG_1P1	Configures Anadig regulator 1.1V unstable event 1 The bitfield is configured as interrupt 0 The bitfield is reset.
23–20 RESERVED	This field is reserved. Reserved for future use
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 RESERVED	This field is reserved. Reserved for future use
5 JTAG_RST	Configures HIGH-Z JTAG event 1 The bitfield is configured as interrupt 0 The bitfield is reset.
4–3 WDOG_RST	Configures watchdog Time-out event. WDOG_RST_F[1] - CM4 core WDOG_RST_F[0] - CA5 core 1 The bitfield is configured as interrupt 0 The bitfield is reset.
2 RESERVED	This field is reserved. Reserved for future use
1 CA5_WDG_RST	Configures event from internal CA5 watchdog timer 1 The bitfield is configured as interrupt 0 The bitfield is reset.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

18.3.7 SRC Interrupt Masking Register (SRC_SIMR)

The SIMR is a interrupt masking register. The event configured in the SCR_SICR as interrupt can be masked in this register. Once the interrupt is masked it will not reflect in SCR_SRSR. Interrupts are masked when the bitfield representing the event is '0'.

Address: 4006_E000h base + 18h offset = 4006_E018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CM4_LCK	SOSC_OK	0	CMU_FLL_FHH	CMU_OLR	REG_3P0_OK	REG_2P5_OK	REG_1P1_OK	RESERVED				0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									RESERVED	JTAG_RST	WDOG_RST		RESERVED	CA5_WDG_RST	0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1

SRC_SIMR field descriptions

Field	Description
31 CM4_LCK	Masks interrupt when CM4 is in lockup 1 Interrupts are non-masked 0 Interrupts are masked
30 SOSC_OK	Masks interrupt no clock is detected on SOSC 1 Interrupts are non-masked 0 Interrupts are masked
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 CMU_FLL_FHH	Masks interrupt CMU event when BUS frequency clock is out of range 1 Interrupts are non-masked 0 Interrupts are masked
27 CMU_OLR	Masks interrupt CMU event of FOSC frequency less than 40 Mhz 1 Interrupts are non-masked 0 Interrupts are masked
26 REG_3P0_OK	Masks interrupt Anadig regulator 3.0V unstable event 1 Interrupts are non-masked 0 Interrupts are masked

Table continues on the next page...

SRC_SIMR field descriptions (continued)

Field	Description
25 REG_2P5_OK	Masks interrupt Anadig regulator 2.5V unstable event 1 Interrupts are non-masked 0 Interrupts are masked
24 REG_1P1_OK	Masks interrupt Anadig regulator 1.1V unstable event 1 Interrupts are non-masked 0 Interrupts are masked
23–20 RESERVED	This field is reserved. Reserved for future use
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
16 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 RESERVED	This field is reserved. Reserved for future use
5 JTAG_RST	Masks interrupt HIGH-Z JTAG event 1 Interrupts are non-masked 0 Interrupts are masked
4–3 WDOG_RST	Masks interrupts watchdog Time-out event. WDOG_RST_F[1] - CM4 core WDOG_RST_F[0] - CA5 core 1 Interrupts are non-masked 0 Interrupts are masked
2 RESERVED	This field is reserved. Reserved for future use
1 CA5_WDG_RST	Masks interrupt event from internal CA5 watchdog timer 1 Interrupts are non-masked 0 Interrupts are masked
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

18.3.8 SRC Boot Mode Register 2 (SRC_SBMR2)

Address: 4006_E000h base + 1Ch offset = 4006_E01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					0	BMOD[1:0]		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								reserved_fuses[2:0]			BT_FUSE_SEL	DIR_BT_DIS	0	SEC_CONFIG[1:0]	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_SBMR2 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 BMOD[1:0]	Please refer to Table 19-1
23–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SRC_SBMR2 field descriptions (continued)

Field	Description
7–5 reserved_ fuses[2:0]	This field is reserved. Please refer to fuse map (connected to hwv_fuse[199:197])
4 BT_FUSE_SEL	Please refer to OCOTP_CFG5[BT_FUSE_SEL] .
3 DIR_BT_DIS	Please refer to OCOTP_CFG5[DIR_BT_DIS] .
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 SEC_ CONFIG[1:0]	Please refer to OCOTP_CFG5[SEC_CONFIG]

18.3.9 General Purpose Register (SRC_GPR*n*)

Address: 4006_E000h base + 20h offset + (4d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GP																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR*n* field descriptions

Field	Description
31–0 GP	Read/write bits, for general purpose.BOOTROM see GPR[1-4] and GPR[9-10] for System Boot. Please refer to Table 19-7 for more details .

18.3.10 MISC0 (SRC_MISC0)

Address: 4006_E000h base + 4Ch offset = 4006_E04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MISC0_31	MISC0_30	MISC0_29	MISC0_28	MISC0_27	MISC0_26	0									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MISC0_15_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_MISC0 field descriptions

Field	Description
31 MISC0_31	For selection between external trigger and usb0 start of frame(sof) for FTM3. Refer to FTM Hardware Triggers. 0 usb0 sof 1 external trigger
30 MISC0_30	For selection between adc1_coco(conversion complete) and usb1 start of frame(sof) for FTM3. 0 adc1_coco 1 usb1 sof
29 MISC0_29	For selection between external trigger and usb0 start of frame(sof) for Flex Timer 2(FTM2). 0 usb0 sof 1 external trigger
28 MISC0_28	For selection between adc1_coco(conversion complete) and usb1 start of frame(sof) for FlexTimer Trigger2. 0 adc1_coco 1 usb1 sof
27 MISC0_27	For selection between external trigger and usb0 start of frame(sof) for FlexTimer Trigger1. 0 usb0 sof 1 external trigger
26 MISC0_26	For selection between external trigger and usb0 start of frame(sof) for FlexTimer Trigger0. For selection between external trigger and usb0 start of frame(sof) for 0 usb0 sof 1 external trigger
25–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 MISC0_15_1	Reserved for future use

18.3.11 MISC1 (SRC_MISC1)

Address: 4006_E000h base + 50h offset = 4006_E050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	MISC1_31	MISC1_30	MISC1_29	MISC1_28	MISC1_27	MISC1_26	MISC1_25_16									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	MISC1_15	MISC1_14	MISC1_13	MISC1_12	MISC1_11	MISC1_10	MISC1_9	MISC1_8	MISC1_7	MISC1_6	MISC1_5	MISC1_4	MISC1_3	MISC1_2	MISC1_1	MISC1_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_MISC1 field descriptions

Field	Description
31 MISC1_31	Selects pre_trigger ->ext_hwts[1] for ADC1 in standby mode. Refer toADC-Digital-12b-1MSPS-SAR(Conversion Control) for more details. **NOTE:** This bit will have effect only if GPC is configured for Standby mode 0 ext_hwts[1] is not selected 1 ext_hwts[1] is selected.
30 MISC1_30	Selects pre_trigger ->ext_hwts[0] for ADC1 in standby mode **NOTE:** This bit will have effect only if GPC is configured for Standby mode. 0 ext_hwts[0] is not selected 1 ext_hwts[0] is selected.
29 MISC1_29	Selects pre_trigger ->ext_hwts[1] for ADC0 in standby mode 0 ext_hwts[1] is not selected 1 ext_hwts[1] is selected.
28 MISC1_28	Selects pre_trigger ->ext_hwts[0] for ADC0 in standby mode 0 ext_hwts[0] is not selected 1 ext_hwts[0] is selected.
27 MISC1_27	Selects between lptimer trigger and PDB trigger for ADC1 0 PDB trigger 1 LPTIMER trigger
26 MISC1_26	Selects between lptimer trigger and PDB trigger for ADC0 0 PDB trigger 1 LPTIMER trigger

Table continues on the next page...

SRC_MISC1 field descriptions (continued)

Field	Description
25–16 MISC1_25_16	Reserved
15 MISC1_15	FTM3 Fault Input3
14 MISC1_14	FTM3 Fault Input2
13 MISC1_13	FTM3 Fault Input1 0 Fault deasserted 1 Fault asserted
12 MISC1_12	FTM3 Fault Input0
11 MISC1_11	FTM2 Fault Input3 0 Fault deasserted 1 Fault asserted
10 MISC1_10	FTM2 Fault Input2 1 Fault deasserted 0 Fault asserted
9 MISC1_9	FTM2 Fault Input1
8 MISC1_8	FTM2 Fault Input0
7 MISC1_7	FTM1 Fault Input3
6 MISC1_6	FTM1 Fault Input2FTM1 Fault Input2
5 MISC1_5	FTM1 Fault Input1
4 MISC1_4	FTM1 Fault Input0
3 MISC1_3	FTM0 Fault Input3
2 MISC1_2	FTM0 Fault Input2
1 MISC1_1	FTM0 Fault Input1
0 MISC1_0	FTM0 Fault Input0

18.3.12 MISC2 (SRC_MISC2)

Address: 4006_E000h base + 54h offset = 4006_E054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	MISC2_31_16															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MISC2_15_10						MISC2_9	MISC2_8	MISC2_7_3					0		MISC2_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_MISC2 field descriptions

Field	Description
31–16 MISC2_31_16	Reserved for future use
15–10 MISC2_15_10	Reserved for future use
9 MISC2_9	Refer Video ADC(VAD). VIU mux select 0 VIU is selected 1 VADC is selected
8 MISC2_8	Anacatum : ext_pd_n 0 VADC disabled 1 VADC enabled
7–3 MISC2_7_3	Reserved
2–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 MISC2_0	DDR RESET control on Standby 0 DDR memory reset is driven by DDR PHY 1 will remain high (inactive)

18.3.13 MISC3 (SRC_MISC3)

Address: 4006_E000h base + 58h offset = 4006_E058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MISC3_15_13			MISC3_12	MISC3_11	MISC3_10	MISC3_9	MISC3_8	0	0	MISC3_5	MISC3_4	MISC3_3	MISC3_2	MISC3_1	MISC3_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_MISC3 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 MISC3_15_13	This field selects the time-out period for the timeout monitor on flexbus. Timeout Period (in Flexbus Clocks) 000 65536 001 32768 010 16384 011 8192 100 4096 101 2048 110 1024 111 512
12 MISC3_12	Enable timeout monitor for Flexbus to prevent device hand on bus error. The timeout value is controlled through SRC_MISCS3[15:13].
11 MISC3_11	Select source of USB1 Overcurrent 0 Drive USB1 OverCurrent from IOMUX 1 Drive USB1 OverCurrent from SRC_MISC3[10].
10 MISC3_10	Drives USB1 OverCurrent when SRC_MISC3[11] .
9 MISC3_9	Select source of USB0 Overcurrent 0 Drive USB0 OverCurrent from IOMUX. 1 Drive USB0 OverCurrent from SRC_MISC3[8].
8 MISC3_8	Drives USB0 overcurrent when SRC_MISC3[9] is asserted.

Table continues on the next page...

SRC_MISC3 field descriptions (continued)

Field	Description
7 Reserved	This read-only field is reserved and always has the value 0.
6 Reserved	This read-only field is reserved and always has the value 0.
5 MISC3_5	QSPI1 - Flash Port A- MACRONIX SEL. Set this bit to configure delays on port corresponding to Macro
4 MISC3_4	QSPI0 - Flash PortB - MACRONIX SEL. Set this bit to configure delays on port corresponding to Macro
3 MISC3_3	QSPI0 - Flash Port A- MACRONIX SEL. Set this bit to configure delays on port corresponding to Macro.
2 MISC3_2	Mask debug_ack of CM4 0 CM4 dbg_ack is Ored with CA5 debug_ack 1 Ignore dbg_ack for CM4
1 MISC3_1	Mask debug_ack of CA5 0 CA5 dbg_ack is Ored with CM4 debug_ack 1 Ignore dbg_ack for CA5
0 MISC3_0	Platform: MLB or ENETM1 select 0 ENET MAC1 as Master9 port 1 MLB as Master9 port

18.4 Functional Description

18.4.1 Reset Sources

The Reset control logic receives event requests from all the potential reset sources. All the "functional Reset/ Interrupt" events are qualified based on the SRC_SICR either as interrupt or reset. All the events are latched in the SRC_SRSR register unless masked by SRC_SIMR register. The reset sources are described in the following table.

Table 18-19. Reset_Table

Reset Source	Reset Type
POR1 - power on reset	Destructive reset
POR1P2: 1.2V supply goes below 0.7V event	Destructive reset
HP LVD: HP regulator's LVD event	Destructive reset
ULP LVD: ULP regulator's LVD event	Destructive reset

Table continues on the next page...

Table 18-19. Reset_Table (continued)

Reset Source	Reset Type
LVD_P3P: 3.3V main supply is unstable	Destructive reset
LP_LVD: LP regulator's LVD event	Destructive reset
CA5 internal WDOG Time-out	Functional Reset / Interrupt
Platform WDOG[CA5 core] Time out	Functional Reset / Interrupt
Platform WDOG[CM4 core] Time out	Functional Reset / Interrupt
JTAGC Reset	Functional Reset / Interrupt
External Reset, RESETB	Functional reset
MDM_AP_CTRL[SYSRESETREQ]	Functional Reset / Interrupt
SNVS_HP: Hard Fail state	Default reset.Generated from SNVS_HP
SRC_SCR[SW_RST]	Functional Reset / Interrupt
CSU Alram reset	Functional Reset / Interrupt
Anadig1.1V regulator unstable	Functional Reset / Interrupt
Anadig 2.5V regulator unstable	Functional Reset / Interrupt
Anadig 3.3V regulator unstable	Functional Reset / Interrupt
CMU: FOSC frequency is < 24MHz	Functional Reset / Interrupt
CMU: Bus frequency clock out of range	Functional Reset / Interrupt
No FOSC clock	Functional Reset / Interrupt
No SOSC clock	Functional Reset / Interrupt
CM4 is in lockup	Functional Reset / Interrupt

- If a destructive reset occurs during a functional reset deassertion sequence, the destructive reset sequence will start without completing the current reset sequence
- If a functional reset occurs during a functional reset deassertion sequence. The reset sequence will will re-initiate
- Reset deassertion sequence won't start until all the resets are deasserted

18.4.2 Destructive reset sequence

The resets from SRC to device are asserted immediately on any destructive reset event. While the clock enables are deasserted at the same time. The RESETB is deasserted to indicate that the device is under reset. The memory repair is initiated only after the CCM indicates that all the clocks are enabled.

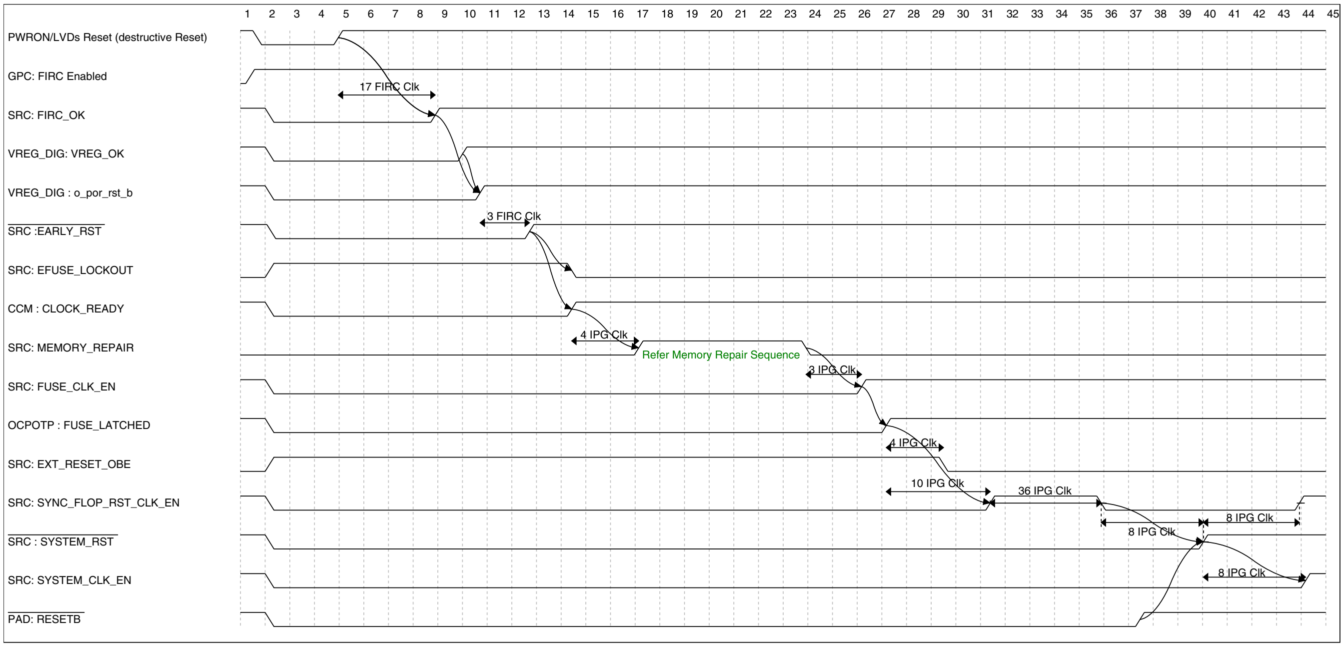


Figure 18-20. Destructive reset sequence

The SRC follows the sequence below on deassertion of poweron or any LVD resets :

1. The stable VREG and FIRC is indicated by VREG by de-asserting the o_por_rst_b.
2. The deassertion of EARLY_RESET wakes the CCM and the OCOTP.
3. Once the CCM indicates that the clocks are stable by asserting the CLOCK_READY the memory repair is initiated.
4. After the memory repair is over the fuses clock is enabled for latching fuses.
5. The latched fuses are decoded by SRC and stored in internal registers.
6. The SYNC_FLOP_RST_CLK_EN is asserted to enable clocks to initialize all non-resetable flops in the device.
7. The deassertion of system reset is held till the external reset RESETB is not released. this is to give controllability to the user on the boot up of the core.
8. The clocks are enabled after the system reset is deasserted, to ensure proper reset recovery

18.4.3 Functional reset sequence

On the assertion of the functional reset the clocks are gated on the edge of the IPG clock to ensure glitch free stop of ongoing transactions. The SRC resets are asserted after the clocks are gated. The RESETB is deasserted to indicate that the device is under reset. The memory repair is not initiated.

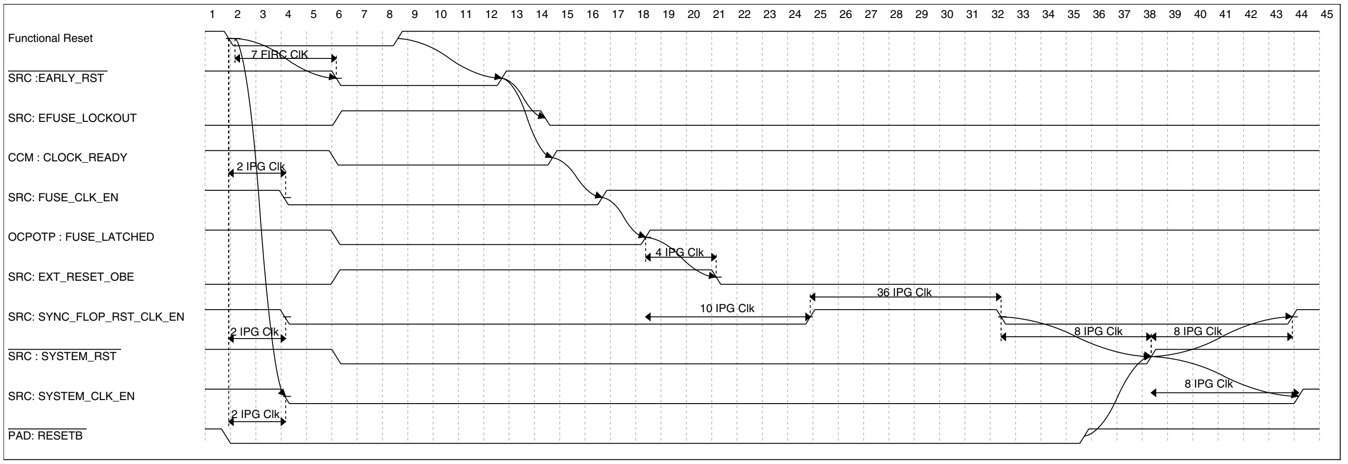


Figure 18-21. Functional reset sequence

The SRC follows the below mentioned sequence on deassertion of any functional reset:

1. The clock enables are deasserted on IPG clock.
2. The SRC resets are asserted after a few FIRC clocks to ensure all clocks are properly gated.
3. The deassertion of EARLY_RESET wakes the CCM and the OCOTP.
4. Once the CCM indicates that the clocks are stable by asserting the CLOCK_READY the clock for the fuse latching is enabled.
5. The latched fuses are decoded by SRC and stored in internal registers.
6. The SYNC_FLOP_RST_CLK_EN is asserted to enable clocks to initialize all non-resetable flops in the device.
7. The deassertion of system reset is held till the external reset RESETB is not released. This is to give controllability to the user on the boot up of the core.
8. The clocks are enabled after the system reset is deasserted, to ensure proper reset recovery.

18.4.4 External reset sequence

On the assertion of the RESETB the clocks are gated on the edge of the IPG clock to ensure glitch free stop of ongoing transactions. The SRC resets are asserted after the clocks are gated. The memory repair is not initiated.

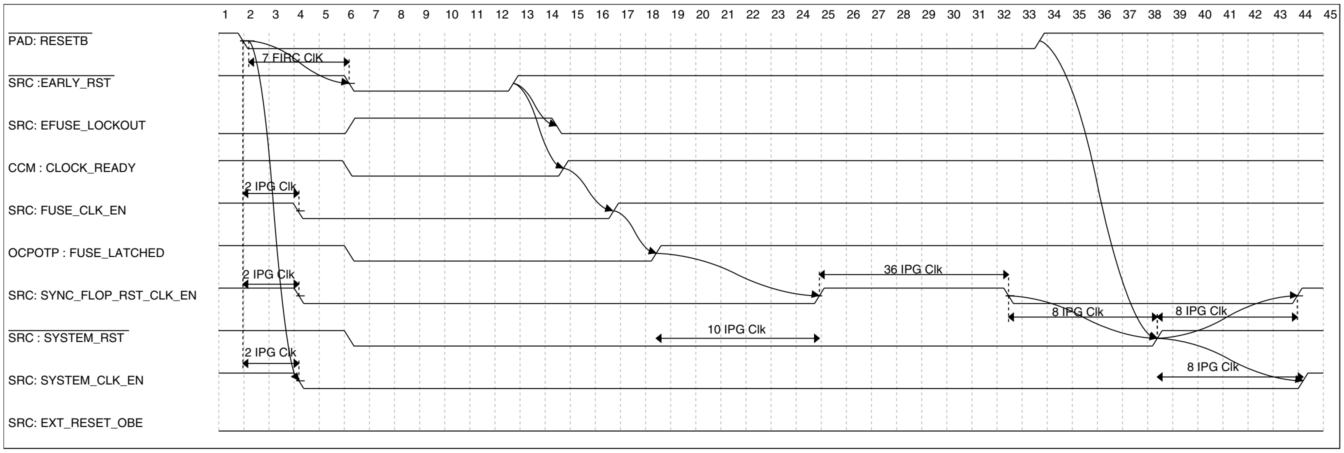


Figure 18-22. External reset sequence

The SRC follows the sequence below on assertion of the external reset, RESETB :

1. The clock enables are deasserted on IPG clock.
2. The SRC resets are asserted after a few FIRC clocks to ensure all clocks are properly gated.
3. The deassertion of EARLY_RESET wakes the CCM and the OCOTP.
4. Once the CCM indicates that the clocks are stable by asserting the CLOCK_READY the clock for the fuse latching is enabled.
5. The latched fuses are decoded by SRC and stored in internal registers.
6. The SYNC_FLOP_RST_CLK_EN is asserted to enable clocks to initialize all non-resetable flops in the device.
7. The deassertion of system reset is held till the external reset RESETB is not released. This is to give controllability to the user on the boot up of the core.
8. The clocks are enabled after the system reset is deasserted, to ensure proper reset recovery.

18.4.5 Standby reset sequence

The Standby reset is asserted after the isolation is activated. On assertion of the standby reset all the clocks are gated and power domain 1 resets are asserted immediately. The memory repair is initiated on exit of Standby.

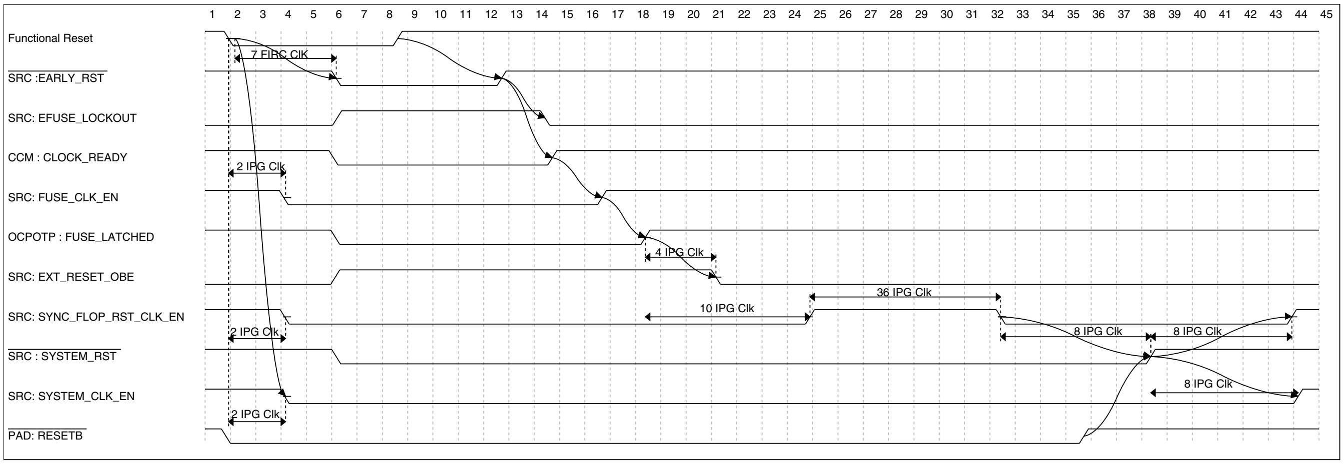


Figure 18-23. Standby reset sequence (Exit by Standby Exit)

The SRC follows the below mentioned sequence on assertion of standby reset :

1. The clocks are gated immediately.
2. The SRC power domain 1 resets are asserted immediately.
3. The deassertion of EARLY_RESET wakes the CCM and the OCOTP
4. Once the CCM indicates that the clocks are stable by asserting the CLOCK_READY, initiating the memory repair sequence.
5. The clock for the fuse latching is enabled after the memory repair.
6. The latched fuses are decoded by SRC and stored in internal registers.
7. The SYNC_FLOP_RST_CLK_EN is asserted to enable clocks to initialize all non-resetable flops in the device.
8. The clocks are enabled after the system reset is deasserted, to ensure proper reset recovery.

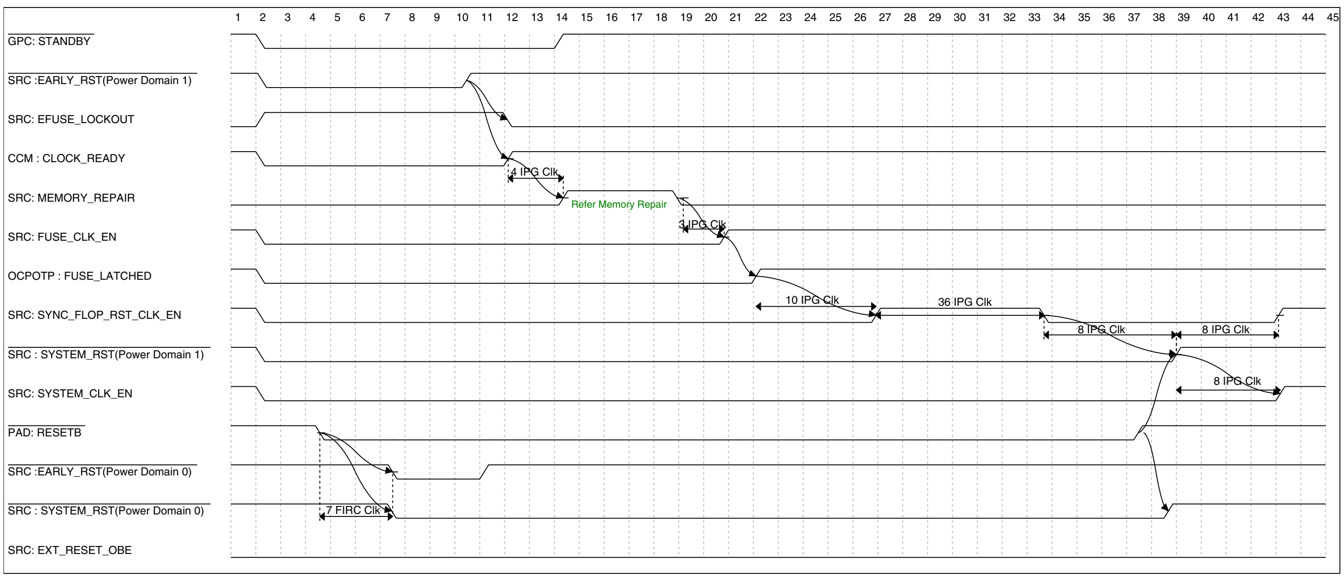


Figure 18-24. Standby reset sequence (Abort by External Reset)

The SRC follows the below mentioned sequence on assertion of standby reset :

1. The clock enables are deasserted immediately.
2. The SRC power domain 1 resets are asserted immediately.
3. The SRC power domain 0 resets are asserted on the assertion of external reset, RESETB. The standby mode is exited.
4. The deassertion of EARLY_RESET wakes the CCM and the OCOTP.
5. Once the CCM indicates that the clocks are stable by asserting the CLOCK_READY, initiating the memory repair sequence.
6. The clock for the fuse latching is enabled after the memory repair.
7. The latched fuses are decoded by SRC and stored in internal registers.
8. The SYNC_FLOP_RST_CLK_EN is asserted to enable clocks to initialize all non-resetable flops in the device.
9. The deassertion of system reset is held till the external reset RESETB is not released. This is to give controllability to the user on the boot up of the core.
10. The clocks are enabled after the system reset is deasserted, to ensure proper reset recovery.

18.4.6 Memory Repair

The memory repair is done after every powerdown of Memory. The two triggering reset events are destructive reset and Standby. The memory repair is initiated after CCM indicates the clocks are stable. The memory repair can be bypassed by assertion signal.

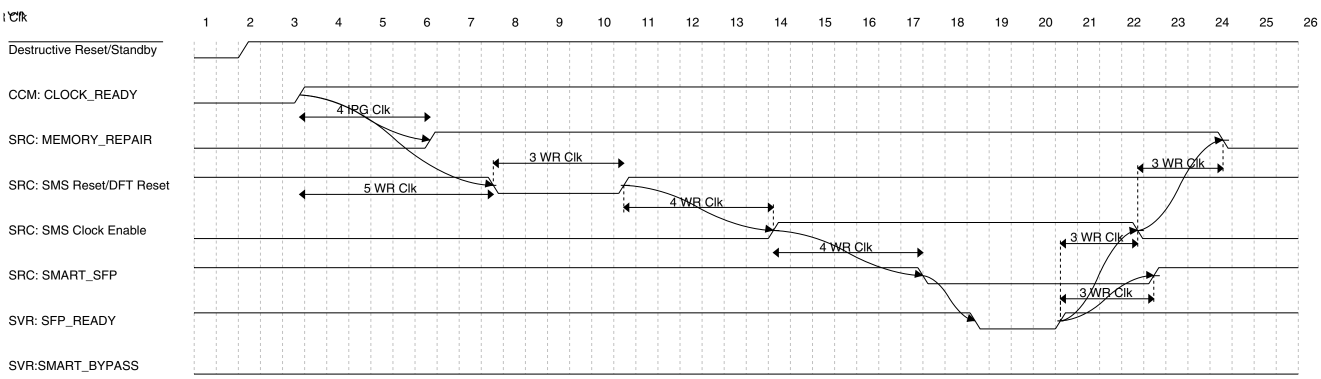


Figure 18-25. Memory Repair Sequence (Bypass ='0')

The memory repair follows the following sequence when SMART_BYPASS ='0' :

1. The memory repair is initiated after the CCM indicates the clocks are ready.
2. The SMS reset are asserted for few memory repair clock cycles.
3. Once the SMS resets are deasserted the memory repair clocks are enabled, for proper clean reset recovery.

4. The SMART_SFP is deasserted.
5. On the assertion of SFP_READY, the SMART_SFP is asserted.
6. The memory repair clock enables are deasserted along with the MEMORY_REPAIR indicator.

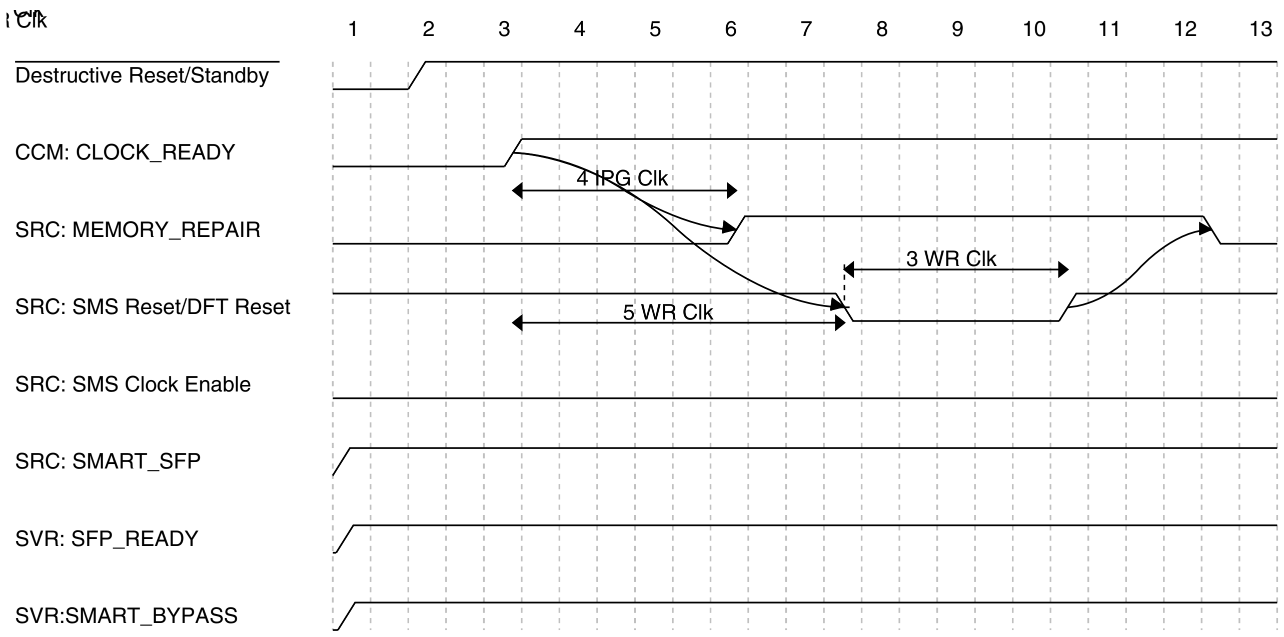


Figure 18-26. Memory Repair Sequence (Bypass ='1')

The memory repair follows following sequence when SMART_BYPASS ='1' :

1. The memory repair is initiated after the CCM indicates the clocks are ready.
2. The SMS reset are asserted for few memory repair clock cycles.
3. The SMS resets are deasserted along with the MEMORY_REPAIR indicator.

18.4.7 BOOTMOD Pin Latching

The exact boot sequence is controlled by the values of the BOOTMOD pins on this device.

All the boot pins will be sampled at deassertion of system_early_rst_b.

The value of the BOOTMOD pins will be latched after the IIM asserts the fuse read completion flag. After latching, the values of the BOOTMOD pins are used to determine the booting options of the core as given in the SBMR register.

The Boot mode general purpose bits can be provided to the SRC from either e-fuses or GPIO signals. The gpio_bt_sel e-fuse defines the source to be used to derive the boot information. When gpio_bt_sel is set, e-fuses are used. When cleared, GPIO signals are used.

When FSL test mode is chosen (11 for BOOTMOD[1:0]) then the GPIO signals are used.

When Internal Production Boot Mode is chosen (00 for BOOTMOD[1:0]) then e-fuses signals are used, independent of the gpio_bt_sel e-fuse value.

The Boot information is provided in SBMR register. The figure below shows the selection of Boot mode information.

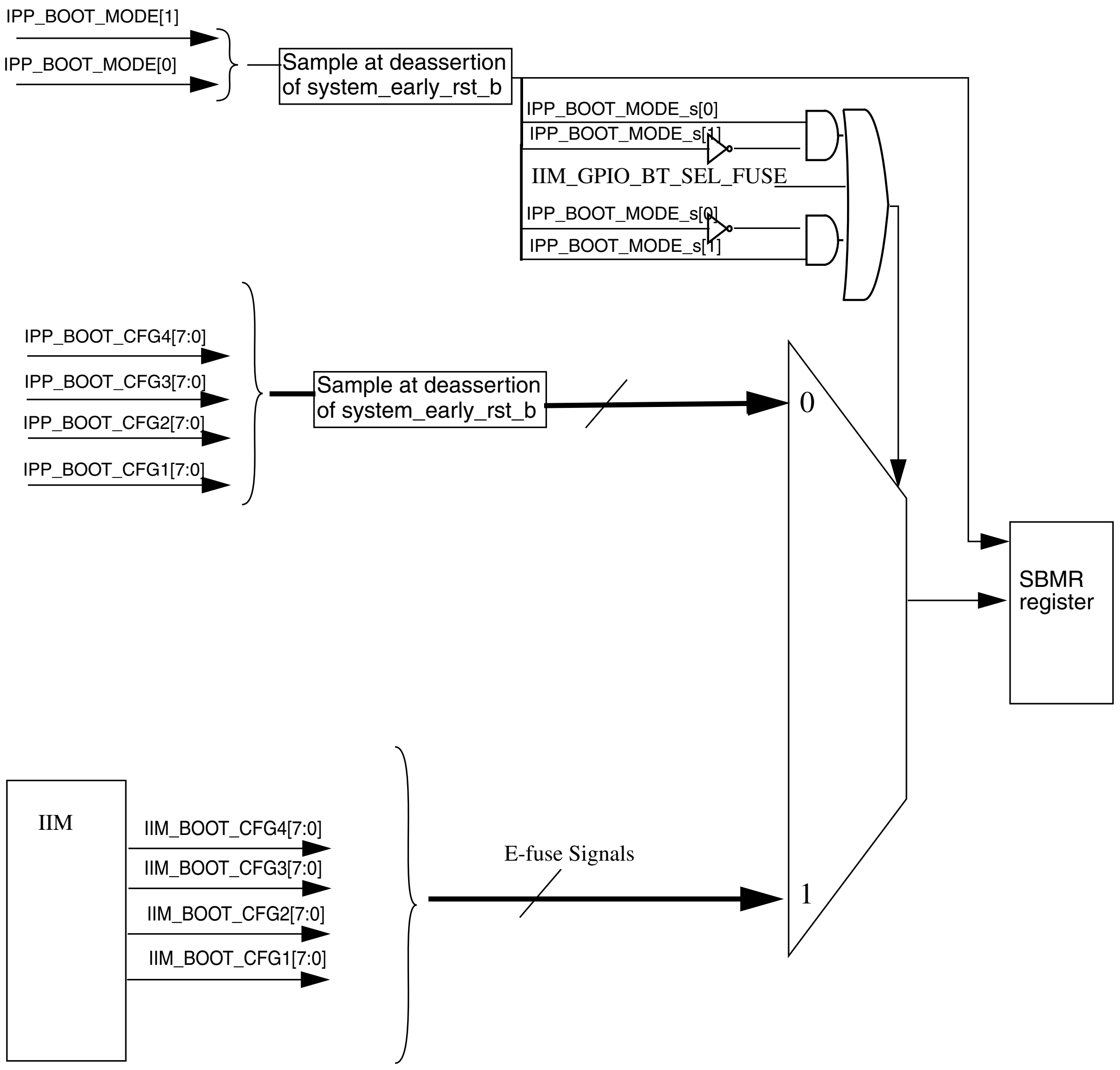


Figure 18-27. Boot Mode Information

Chapter 19
System Boot

19.1 Introduction

The boot process begins at Power On Reset (POR) where the hardware reset logic forces the ARM core to begin execution starting from the on-chip Boot ROM. The Boot ROM code uses the state of the internal register BOOT_MODE [1:0] as well as the state of various eFUSEs and/or GPIO settings to determine the boot flow behavior of the device.

The main features of the ROM include:

- Support for booting from various boot devices
- Serial Downloader support (USB and UART)
- Device Configuration Data (DCD)
- Wakeup from low-power modes
- Wakeup secondary core
- HAB Re-authentication on Low Power Standby exit

Boot ROM supports the following boot devices:

- NOR Flash
- NAND Flash
- FlexCAN
- SD/MMC
- QuadSPI
- Serial ROM devices (through I2C and SPI interfaces)

The boot process can be summarized in the following steps.

1. Reset.
2. Begin execution of Boot ROM code. The boot process depends on the reset type, HAB_REAUTH setting, CPU ID, BOOT_MODE[1:0], and the RCON register.
3. Read the Image Vector Table and boot data structures. Optionally, download image to memory.
4. Execute image, i.e., the user code.

In normal operation, the Boot ROM uses the state of the BOOT_MODE register and eFUSEs to determine the boot device. For development purposes, there is also an option to use GPIO pin inputs to override the fuse values to determine the boot device.

NOTE

BOOT_MODE and and GPIO Pins are sampled on a Power On Reset event.

The Boot ROM code also allows for programs to be downloaded and executed on the device. An example is a provisioning program that can make further use of the serial connection to program a boot device with a new image. Typically, the provisioning program is downloaded to internal RAM and programs the boot devices such as a NAND Flash. The ROM Serial Downloader can use either the high speed USB or UART interface in non-stream mode connection.

The Boot ROM allows waking from low power modes and waking up secondary core (either Cortex-A5 or Cortex-M4 depending on configuration). On reset the Boot ROM checks the CPU ID and power gating status register. The primary core (with ID = 0), on waking up from low power mode, and, secondary core (with ID=1) in all power modes, will skip downloading the image from the boot device. It will jump to the address saved in the PERSISTENT_ENTRY Persistent Bits register.

The Device Configuration Data (DCD) feature allows the Boot ROM code to obtain system configuration data from an external Program Image residing on the boot device. As an example, the DCD can be used to program the SDRAM controller for optimal settings to improve the boot performance. DCD configuration options are restricted to memory areas and peripheral addresses that are considered essential for boot purposes.

The remainder of this chapter provides the details on how to configure and use the boot features of the device.

19.2 Boot Modes

On reset, the device checks the CPU ID in the MSCM CPxNUM register and Power Gating Controller status register.

On normal boot, the Primary Core behavior is defined by the Boot Mode pins settings as described in Boot Mode Pin Settings. On waking up from low power boot mode, the Primary Core skips configuring the clocks. The Boot ROM checks that PERSISTENT_ENTRY0 (see Persistent Bits) is a pointer to a valid address space (internal SRAM, DDR, FlexBus, and QuadSPI). If PERSISTENT_ENTRY0 is a pointer to a valid range, it starts execution using the entry point from the PERSISTENT_ENTRY0 register. If PERSISTENT_ENTRY0 is a pointer to an invalid range, the Primary Core performs a system reset.

Note

Wake-up from Internal SRAM is only supported as clocks to other modules are disabled.

For the Secondary Core, the Boot ROM checks that the PERSISTENT_ENTRY is a pointer to a valid address space (Internal SRAM, DDR, FlexBus, and QuadSPI). If PERSISTENT_ENTRY is a pointer to a valid range, it starts execution using the entry point from the PERSISTENT_ENTRY register. If PERSISTENT_ENTRY is a pointer to invalid range, it sets error status registers (see Persistent Bits), sends a wakeup error interrupt, and performs a Wait-For-Interrupt instruction. The Interrupt Service routine of primary core must reconfigure the system and reset the secondary core.

Note

SW that enables secondary core is not part of ROM. It must be part of upper level SW.

19.2.1 Boot Mode Pin Settings

The device has four boot modes (one is reserved for Freescale use). Boot mode is selected based on the binary value stored in the internal BOOT_MODE register. BOOT_MODE is initialized by sampling the BOOT_MODE0 and BOOT_MODE1 pins on the rising edge of RESET_B after POR. After these pins are sampled, their subsequent state does not affect the contents of the BOOT_MODE internal register. The state of the internal BOOT_MODE register may be read from the BMOD[1:0] field of the SRC Boot Mode Register (SRC_SBMR2). The available boot modes are: Boot From Fuses, serial boot via USB or UART, Boot from RCON and FSL Test mode. See the table below for settings.

Table 19-1. Boot MODE Pin Settings

BOOT_MODE[1:0]	Boot Type
00	Boot From Fuses

Table continues on the next page...

Table 19-1. Boot MODE Pin Settings (continued)

BOOT_MODE[1:0]	Boot Type
01	Serial Downloader
10	Boot from RCON

19.2.2 High Level Boot Sequence

The figure below shows the High Level boot ROM code flow in this device.

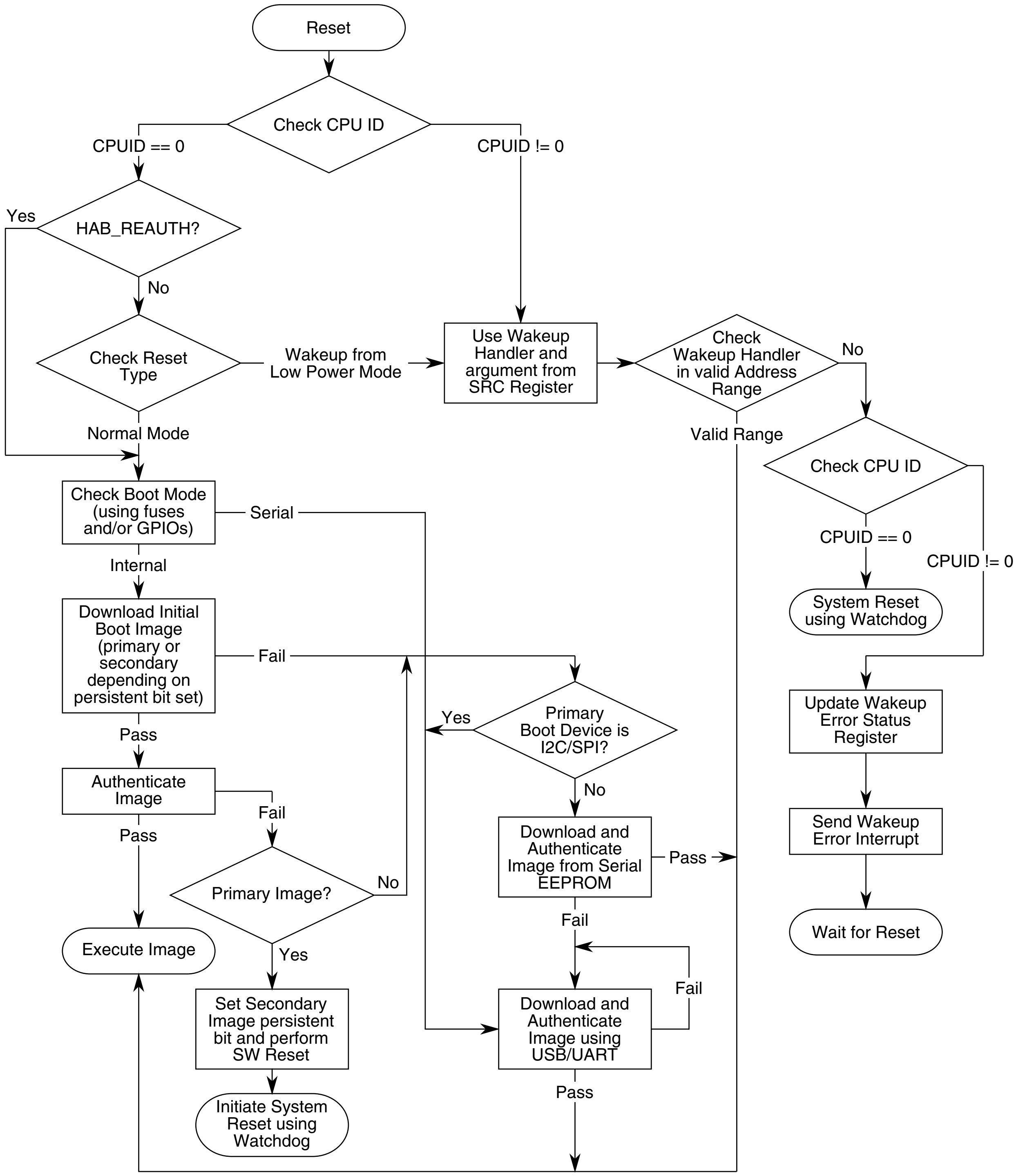


Figure 19-1. BOOT Flow

Note

For FlexBus (NOR) and QuadSPI boot devices, downloading initial load region to iRAM is skipped. IVT is read from FlexBus/QuadSPI address space respectively (see Image Vector

Table and Boot Data). Copying initial load region and the rest of the program image is done only if the absolute start address of the image is not equal to FlexBus/QuadSPI start address.

19.2.3 Boot From Fuses Mode (BOOT_MODE [1:0] = 0b00)

A value of 0b00 in the BOOT_MODE [1:0] register selects the Boot From Fuses mode. This mode is similar to the Boot from RCON mode, with one difference. In this mode the GPIO boot override pins are ignored. The boot ROM code uses the boot eFUSE settings only. This mode also supports a secure boot using HAB.

If set to Boot From Fuses, the boot flow is controlled by the BT_FUSE_SEL eFUSE value. If BT_FUSE_SEL = 0, indicating that the boot device (for example, Flash, SD/MMC) has not yet been programmed, the boot flow jumps directly to the Serial Downloader. If BT_FUSE_SEL = 1, then the normal boot flow is followed, where the ROM attempts to boot from the selected boot device.

The first time a board is used, the default eFUSEs may be configured incorrectly for the hardware on the platform. In such a case, the Boot ROM code may try to boot from a device that does not exist. This may cause an electrical/logic violation on some pads. Using Boot From Fuses mode solves this problem. The first time the BT_FUSE_SEL eFuse is encountered, it is not blown (thus BT_FUSE_SEL=0). This forces the ROM code to jump directly to the Serial Downloader. This allows a bootloader to be downloaded, which can then provision the boot device with a Program Image, and blow the BT_FUSE_SEL and the other boot configuration eFUSEs. After reset, the Boot ROM code determines that BT_FUSE_SEL is blown (BT_FUSE_SEL = 1) and the ROM code performs internal boot according to the new eFUSE settings. This allows a user to set BOOT_MODE [1:0] =0b00 on a production device and burn fuses on the same device (by forcing entry to the Serial Downloader), without changing the value of BOOT_MODE [1:0] or pull-ups/pull-downs on the BOOT_MODE pins.

19.2.4 Mode: Serial Downloader Mode (BOOT_MODE [1:0] = 0b01)

The serial downloader is invoked if the external Flash device is not programmed, when a failure is encountered during the boot flow process, or any of the following conditions are met:

- BOOT_MODE[1:0] = 0b01 (serial downloader mode)
- BOOT_MODE[1:0] = 0b00 (boot from fuses) and the eFUSE BT_FUSE_SEL = 0

- BOOT_MODE[1:0] = 0b00 or 0b10 (Boot From Fuses or Boot From RCON) and there is not a valid image in the external boot
- Internal failure
- Runtime exception occurs
- Error is returned by the HAB functions while in a Closed security configuration. Errors are ignored in an Open security configuration (see Boot Security Settings).

To determine the USB/UART connection, the Boot ROM polls the USB/UART status register.

If WDOG_ENABLE eFUSE is 1 and there is no activity on USB/UART port within the predefined polling time, then the ROM program will power down the SoC using the Watchdog Timer (either WDOG-A5 or WDOG-M4 depending on the primary core). When the serial bootloader is active, the Watchdog is serviced periodically. If the communication between the serial Host and this device is idle for more than 90 seconds, or the processor goes into an endless loop, then the watchdog expires and powers down the device. For details on the Serial Downloader see Serial Downloader (BOOT_MODE [1:0] = 0b01).

19.2.5 Boot from RCON (BOOT_MODE [1:0] = 0b10)

A value of 10 in the BOOT_MODE [1:0] register selects the Boot from RCON. In this mode, the processor continues to execute boot code from the internal boot ROM. The boot code performs hardware initialization, loads the Program Image from the chosen boot device, performs image validation using the HAB library (see Boot Security Settings), and then jumps to an address derived from the Program Image. If any error occurs during internal boot, the boot code jumps to the Serial Downloader (see Serial Downloader (BOOT_MODE [1:0] = 0b01)). A secure boot using the HAB is possible in all the three boot modes.

When set to Boot from RCON, the boot flow may be controlled by a combination of eFUSE settings with an option of overriding the fuse settings using GPIO General Purpose I/O (GPIO) pins. Whether the ROM uses GPIO pins for a select number of configuration parameters or eFUSEs in this mode is determined by the GPIO Boot Select (BT_FUSE_SEL) eFUSE.

- If BT_FUSE_SEL = 1, all boot options are controlled by the eFUSEs described in Table 19-2.
- If BT_FUSE_SEL = 0, specific boot configuration parameters may be set using GPIO pins rather than eFUSEs. The fuses that can be overridden when in this mode are indicated in the GPIO column of Table 19-2. Table 19-3 provides the details on the GPIO pins.

The use of GPIO overrides is intended for development since these pads are used for other purposes in deployed products. Freescale recommends controlling the boot configuration by eFUSEs in deployed products and reserving the use of the GPIO mode for development and testing purposes only.

19.3 Device Configuration

This section describes the external inputs that control the behavior of the Boot ROM code. This includes boot device selection (NAND, NOR, MMC), boot device configuration (NAND address cycles), and so on. In general, the source for this configuration comes from eFUSEs embedded in this device. However, certain configuration parameters can be sourced from GPIO pins allowing further flexibility during the development process.

19.3.1 Boot eFUSE Descriptions

The table below is a comprehensive list of the configuration parameters that the device ROM uses.

Table 19-2. BOOT eFuse Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings[2]
DIR_BT_DIS	OEM	Reserved FSL Test/ WDOG-1 Reset boot Mode Disable	NA	0	0 Reserved FSL Test/Watchdog Reset boot Mode is allowed 1 Reserved FSL Test/ Watchdog Reset boot Mode is not allowed

Table continues on the next page...

Table 19-2. BOOT eFuse Descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings[2]
BT_FUSE_SEL	OEM	In Boot from RCON, the BT_FUSE_SEL fuse determines whether the boot settings indicated by a Yes in the GPIO column are controlled by either GPIO pins or eFUSE settings in the IC Identification Module (IIM). In Boot From Fuses mode, the BT_FUSE_SEL determines whether device goes directly into Serial Bootloader mode, or if the boot settings are determined by eFuses.	NA	0	If BOOT_MODE[1:0] = 0b10 0 Bits of SBMR are overridden by GPIO pins. 1 Specific bits of SBMR are controlled by eFUSE settings. If BOOT_MODE[1:0] = 0b00 0 BOOT configuration eFUSEs are not yet programmed. Boot flow jumps to serial downloader. 1 BOOT configuration eFUSEs have been programmed. Regular boot flow is performed.
DIE-XCORDINATE[7:0] DIE-YCORDINATE[7:0] WAFER_NO[4:0] LOT_NO_ENC[42:40] LOT_NO_ENC[39:32] LOT_NO_ENC[31:24] LOT_NO_ENC[23:16] LOT_NO_ENC[15:8] LOT_NO_ENC[7:0]	FSL	Device Unique ID, 64-bit UID.	NA	Unique ID	Settings vary - used by HAB
BT_MMU_DISABLE	OEM	MMU/L1 D Cache/L2 Cache disable bit used by boot ROM for fast HAB processing	NA	0	For Cortex-A5 primary boot: 0 MMU/L1 D Cache/L2 Cache is enabled by ROM during the boot 1 MMU/L1 D Cache/L2 Cache is disabled by ROM during the boot For Cortex-M4 as primary boot: • 0 - L1 D Cache is enabled by ROM during the boot • 1 - L1 D Cache is disabled by ROM during the boot
BOOT_CFG1[7:0]	OEM	Boot Configuration 1	Yes	0	Specific to selected boot mode. See the attached "Fuse-RCON Mapping.xlsx" for details.
BOOT_CFG2[7:0]	OEM	Boot Configuration 2	Yes	0	Specific to selected boot mode. See the attached "Fuse-RCON Mapping.xlsx" for details.

Table continues on the next page...

Table 19-2. BOOT eFuse Descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings[2]
BOOT_CFG3[7:0]	OEM	Boot Configuration 3	Yes	0	Specific to selected boot mode. See the attached "Fuse-RCON Mapping.xlsx" for details.
BOOT_CFG4[7:0]	OEM	Boot Configuration 4	Yes	0	Specific to selected boot mode. See the attached "Fuse-RCON Mapping.xlsx" for details.
WDOG_ENABLE	OEM	Watchdog Reset Counter enable	No	0	0 - watchdog reset counter is disabled 1 - watchdog reset counter is enabled
BOOT_CFG4[7]/RCON31	OEM	If is 1, Boot will wait in an infinite loop. This is only for Debug purpose.	Yes	0	0 is Disabled 1 is Enabled
BOOT_CFG3[0]/RCON16	OEM	Reserved	Yes	0	Should be set to zero
OSC_BYPASS[0][3]	OEM	Controls oscillator bypass	No	0	0 - Oscillator Bypass Disabled 1 - Oscillator Bypass Enabled
OSC_TUNE D[7:0][4]	OEM	Used to change the reset value of CCM_CCR[OSCNT].	No	0	BOOTROM programs CCM_CCR[OSCNT] 00 - CCM_CCR[OSCNT] contains reset value. XX - CCM_CCR[OSCNT] is programmed with value of OSC_TUNED

1. Setting can be overridden by GPIO settings (on a POR event) when BT_FUSE_SEL fuse is intact. See Table 19-3 for corresponding GPIO pin.
2. 0 = intact fuse and 1= blown fuse
3. Bank0 Word 7 Bit 16
4. Bank0 Word7 Field[31:24]

19.3.2 GPIO Boot Overrides

The table below provides a list of GPIO boot overrides. These input pins are sampled at POR, and can be used to override corresponding eFUSE values, depending on the setting of the BT_FUSE_SEL fuse. The GPIO override feature is only available when BT_FUSE_SEL is 0 (fuse is un-blown) and BOOT_MODE [1:0] = 10.

Table 19-3. GPIO Override Contact Assignments

Port Name	RCON Number	SRC_SBMR
PTE1	BMODE[0]	BMOD[0]
PTE0	BMODE[1]	BMOD[1]
PTE7	RCON0	BOOT_CFG1[0]

Table continues on the next page...

Table 19-3. GPIO Override Contact Assignments (continued)

Port Name	RCON Number	SRC_SBMR
PTE8	RCON1	BOOT_CFG1[1]
PTE9	RCON2	BOOT_CFG1[2]
PTE10	RCON3	BOOT_CFG1[3]
PTE11	RCON4	BOOT_CFG1[4]
PTE12	RCON5	BOOT_CFG1[5]
PTE15	RCON6	BOOT_CFG1[6]
PTE16	RCON7	BOOT_CFG1[7]
PTE17	RCON8	BOOT_CFG2[0]
PTE18	RCON9	BOOT_CFG2[1]
PTE19	RCON10	BOOT_CFG2[2]
PTE20	RCON11	BOOT_CFG2[3]
PTE23	RCON12	BOOT_CFG2[4]
PTE24	RCON13	BOOT_CFG2[5]
PTE25	RCON14	BOOT_CFG2[6]
PTE26	RCON15	BOOT_CFG2[7]
PTE27	RCON16	BOOT_CFG3[0]
PTE28	RCON17	BOOT_CFG3[1]
PTC0	RCON18	BOOT_CFG3[2]
PTC1	RCON19	BOOT_CFG3[3]
PTC2	RCON20	BOOT_CFG3[4]
PTB26	RCON21	BOOT_CFG3[5]
PTB27	RCON22	BOOT_CFG3[6]
PTB28	RCON23	BOOT_CFG3[7]
PTC26	RCON24	BOOT_CFG4[0]
PTC27	RCON25	BOOT_CFG4[1]
PTC28	RCON26	BOOT_CFG4[2]
PTC29	RCON27	BOOT_CFG4[3]
PTC30	RCON28	BOOT_CFG4[4]
PTC31	RCON29	BOOT_CFG4[5]
PTB1	RCON30	BOOT_CFG4[6]
PTB2	RCON31	BOOT_CFG4[7]

19.3.3 Device Configuration Data

See Device Configuration Data (DCD) for more details on Device Configuration Data.

19.4 Device Initialization

This section describes the details on the device ROM and provides initialization details. This includes details on the following:

- The iROM Memory Map
- The iRAM Memory Map
- On-chip blocks that the ROM should make use of or change POR register default values
- Clock initialization
- Enabling the MMU/L2 cache when performing a secure boot (SEC_CONFIG = Closed)
- Exception handling, error logging, and interrupt handling

19.4.1 Internal ROM / RAM Memory Map

The figure below shows the internal ROM and internal RAM memory map for the device.

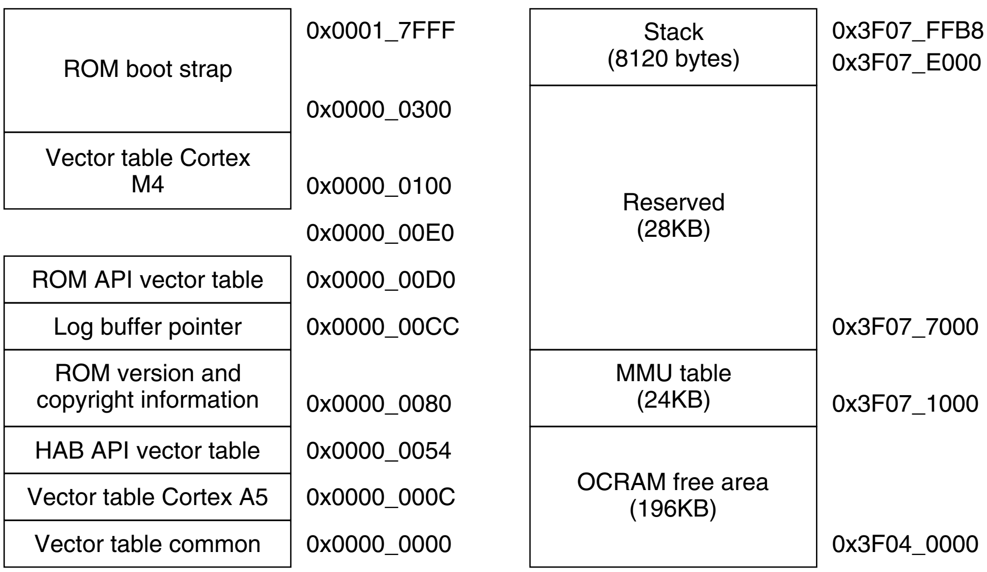


Figure 19-2. Internal ROM and RAM Memory Map

NOTE

RAM Areas will be overwritten by the BOOTROM.
Application can use this area after the control is transferred.

19.4.2 Boot Block Activation

The device Boot ROM affects a number of different hardware blocks which are activated and play a vital role in the boot flow. The ROM configures and uses the following blocks during the boot process. Note that the blocks actually used depend on the boot mode and boot device selection:

- ANADIG - Analog module. The ANADIG contains PLLs and USB PHY.
- MSCM — Miscellaneous System Configuration Module
- SRC - System Reset Controller
- CCM - Clock Control Module
- OCOTP - On-Chip OTP Controller. The OCOTP contains the eFUSEs.
- GPIO — GPIO Controller
- PIT — Programmable Interrupt Timer
- IOMUXC - I/O Multiplexer Control allows GPIO use to override eFUSE boot settings GPIO - used to choose tests in FSL Test Mode
- SPI — Serial Peripheral Interface
- FlexBus Controller - External Memory Interface. Used for NOR devices
- QuadSPI — Quad SPI controller
- I2C - Inter IC Controller
- NFC - NAND Flash controller pin interface
- USB - Used for serial download of a boot device provisioning program
- ESDHC — Enhanced Secure Digital Host Controller
- FlexCAN — CAN Controller
- UART — To Support Serial Boot
- WDOG-A5 - Watchdog Cortex-A5 Timer
- WDOG-M4 — Watchdog Cortex-M4 Timer

19.4.3 Clocks at Boot Time

Boot ROM provides facility to bypass external oscillator. This can be used by blowing OSC_BYPASS efuse. Refer to Boot eFUSE Descriptions

Boot ROM also allows customer to override the reset value of CCM_CCR[OSCNT] field by programming OSC_TUNED eFuse (Refer to Boot eFUSE Descriptions) when OSC_BYPASS eFuse is not blown. This can be helpful when the crystal startup time is more. Boot ROM performs a timed poll for CCM_CSR[FXOSC_RDY] to be asserted. The time is computed by the formula: Wait Time = 1.5*(CCM_CCR[OSCNT]/SIRC) if CCM_CSR[FXOSC_RDY] is not set within the time computed, then BOOTROM enters fail safe mode (serial download from UART).

The table below shows the various clocks and their sources used by ROM.

Table 19-4. Normal Frequency Clocks Configuration

Clock	CCM Signal	Source	Frequency(MHz)
Cortex-A5 Clock	ca5_mux_out_clk	PLL1_PFD4	264
Cortex-M4 Clock (Platform Bus Clock)	cm4_div_out_clk	PLL1_PFD4	132
IPG Clock	Bus_div_out_clk	PLL1_PFD4	66
FlexBUS Clock		Platform Bus Clock	33 (Configurable by Application)
NFC Clock	nfc_clk_root	PLL3_PFD1 PLL1_PFD2	30.85 (Normal Frequency) 45.25 (Fast Frequency)
QSPI0 Clock	qspi0_serial_b_clk qspi0_serial_clk qspi0_serialx2_b_clk qspi0_serialx2_clk qspi0_serialx4_b_clk qspi0_serialx4_clk	PLL3_PFD4 PLL3_MAIN PLL3_PFD4 PLL3_PFD4	18.62 (INIT) 60 74.5 99.3(SDR)
QSPI1 Clock	qspi1_serial_b_clk qspi1_serial_clk qspi1_serialx2_b_clk qspi1_serialx2_clk qspi1_serialx4_b_clk qspi1_serialx4_clk	PLL3_PFD4 PLL3_MAIN PLL3_PFD4 PLL3_PFD4	18.62 (INIT) 60 74.5 99.3(SDR)
eSDHC0 Clock	esdhc0_clk_root	Platform Bus Clock PLL1_PFD3 PLL1_PFD3 PLL3_PFD3 PLL3_PFD3	400KHz (INIT) 24.75 MHz (SD Operating) 49.5 MHz (SD High) 18.62 MHz(MMC Operating) 42.5 MHz (MMC High)

Table continues on the next page...

Table 19-4. Normal Frequency Clocks Configuration (continued)

Clock	CCM Signal	Source	Frequency(MHz)
eSDHC1 Clock	esdhc1_clk_root	Platform Bus Clock PLL1_PFD3 PLL1_PFD3 PLL3_PFD3 PLL3_PFD3	400KHz (INIT) 24.75MHz (SD Operating) 49.5MHz (SD High) 18.62MHz(MMC Operating) 42.5MHz (MMC High)
USB Clock	usb_clk		
CAN0 Bus Clock		External Oscillator Clock	1 MHz
CAN1 Bus Clock		External Oscillator Clock	1 MHz
I2C0 Bus Clock		IPG Clock	353 KHz (ipg_clk = 24MHz) 342 KHz (ipg_clk = 66MHz)
I2C1 Bus Clock		IPG Clock	353KHz (ipg_clk = 24MHz)
I2C2 Bus Clock		IPG Clock	342KHz (ipg_clk = 66MHz)
I2C3 Bus Clock		IPG Clock	353KHz (ipg_clk = 24MHz)
SPI0 Bus Clock		IPG Clock	EEPROM --~5MHz FLASH - ~20MHz
SPI1 Bus Clock		IPG Clock	EEPROM ~5MHz FLASH - ~20MHz
SPI2 Bus Clock		IPG Clock	EEPROM ~5MHz FLASH - ~20MHz
SPI3 Bus Clock		IPG Clock	EEPROM ~5MHz FLASH - ~20MHz
UART0 Baud Rate		IPG Clock	115200bps
UART1 Baud Rate		IPG Clock	115200bps
UART2 Baud Rate		IPG Clock	115200bps
UART3 Baud Rate		IPG Clock	115200bps

On reset the processor has access to all peripherals. ROM code will disable the clocks listed in Table 19-5, except for the boot devices listed in the second column.

Table 19-5. List of Disabled Clocks

Clock Name	Enabled for BOOT Device
CCGR0_FLEXCAN_0	FlexCAN0
CCGR0_DMA0_CH_MUX_0	
CCGR0_DMA0_CH_MUX_1	
CCGR0_UART0	UART0
CCGR0_UART1	UART1
CCGR0_UART2	UART2
CCGR0_UART3	UART3

Table continues on the next page...

Table 19-5. List of Disabled Clocks (continued)

Clock Name	Enabled for BOOT Device
CCGR0_SPI0	SPI0
CCGR0_SPI1	SPI1
CCGR0_SAI0	
CCGR1_SAI1	
CCGR1_SAI2	
CCGR1_SAI3	
CCGR1_CRC	CRC (enabled by HAB)
CCGR1_USB_OTG	USB
CCGR1_PDB	
CCGR1_PIT	PIT (enabled by HAB)
CCGR1_FTM0	
CCGR1_FTM1	
CCGR1_ADC0	
CCGR1_TCON0	
CCGR1_WDOG_A5	Cortex-A5 Watchdog (enabled for Cortex-A5 Primary Core)
CCGR1_WDOG_M4	Cortex-M4 Watchdog (enabled for Cortex-M4 Primary Core)
CCGR2_LPTMR	
CCGR2_RLE	
CCGR2_MLB	
CCGR2_QSPI0	Quad SPI0
CCGR2_IOMUXC	IOMUX (enabled by HAB)
CCGR2_GPIO_PORT0_CTRL	
CCGR2_GPIO_PORT1_CTRL	
CCGR2_GPIO_PORT2_CTRL	
CCGR2_GPIO_PORT3_CTRL	
CCGR2_GPIO_PORT4_CTRL	
CCGR3_ANADIG	ANADIG (enabled by HAB)
CCGR3_SCSC	Slow Clock Source Control (enabled by HAB)
CCGR3_DCU0	
CCGR4_ASRC	
CCGR4_SPDIF	
CCGR4_ESAI	
CCGR4_ESAI_BIFIFO	
CCGR4_EWM	
CCGR4_I2C0	I2C0
CCGR4_I2C1	I2C1
CCGR4_WKUP	
CCGR4_CCM	CCM (enabled by HAB)
CCGR4_GPC	Global Power Controller (enabled by HAB)

Table continues on the next page...

Table 19-5. List of Disabled Clocks (continued)

Clock Name	Enabled for BOOT Device
CCGR4_VREG_DIG	
CCGR4_SRC	System Reset Controller (enabled by HAB)
CCGR4_CMU	
CCGR6_DMA1_CH_MUX_0	
CCGR6_DMA1_CH_MUX_1	
CCGR6_SJTAG	Secure JTAG (enabled by HAB)
CCGR6_OTP_CTRL	OCOTP Controller (enabled by HAB)
CCGR6_SNVS	SNVS (enabled by HAB)
CCGR6_WDOG_SNVS	WDOG — SNVS (enabled by HAB)
CCGR6_UART4	
CCGR6_UART5	
CCGR6_SPI2	SPI2
CCGR6_SPI3	SPI3
CCGR6_DDRMC	
CCGR7_SDHC0	eSDHC0
CCGR7_SDHC1	eSDHC1
CCGR7_USB_OTG2	
CCGR7_FTM2	
CCGR7_FTM3	
CCGR7_ADC1	
CCGR7_TCON1	
CCGR7_SLCD	
CCGR8_QSPI1	Quad SPI1
CCGR8_VADC	
CCGR8_VDEC	
CCGR8_VIU	
CCGR8_DAC0	
CCGR8_DAC1	
CCGR8_OPEN_VG	
CCGR9_ENET0	
CCGR9_ENET1	
CCGR9_FLEXCAN_1	FlexCAN 1
CCGR9_DCU1	
CCGR9_NFC	Nand Flash Controller
CCGR10_I2C2	I2C2
CCGR10_I2C3	I2C3
CCGR10_ENET_L2_SWITCH	
Reserved	Reserved
CCGR11_GPIOC	GPIO Controller (enabled by HAB)

Table continues on the next page...

Table 19-5. List of Disabled Clocks (continued)

Clock Name	Enabled for BOOT Device
CCPGR1_FLEXBUS	FlexBus

19.4.4 Enabling MMU and Caches

The Boot ROM includes a feature of enabling the Memory Management Unit (MMU) and caches to improve the boot speed when performing a secure boot with SEC_CONFIG = Closed (High Assurance Boot (HAB)). L1 data cache, L2 cache and MMU are enabled during image authentication.

By default L1 I-Cache is enabled.

The MMU feature is controlled by the BT_MMU_DISABLE eFUSE. By default the BT_MMU_DISABLE eFUSE is not blown meaning the ROM uses MMU, L1 Data and L2 caches of the Cortex-A5 core. The ROM establishes a page table in the MMU Page Table area of iRAM. The ROM also configures the L1 and L2 cache as write through. This improves the performance of the HAB signature verification software.

During normal boot , enabling the MMU and setting the CSF pointer in the Image Vector Table to NULL has no impact on the boot performance. When SEC_CONFIG=Open is the final configuration, it is recommended to blow the BT_MMU_DISABLE fuse. The Table below describes the relationship between BT_MMU_DISABLE and L2_CACHE_DISABLE fuse bits and its impact on BOOTROM implementation for Cortex-A5 and Cortex-M4 cores:

Table 19-6. Impact of BT_MMU_DISABLE & L2_CACHE_DISABLE on BOOTROM Implementation

Booting Core	BT_MMU_DISABLE	L2_CACHE_DISABLE	Remarks
Cortex-A5	0	0	L1-DCache, MMU & L2 Cache enabled
	0	1	L1-DCache, MMU Enabled & L2 Cache disabled
	1	X	L1-Dcache, MMU & L2 Cache Disabled
Cortex-M4	X	X	L1-DCache enabled

Note

If the Cortex-M4 is the primary core, the MMU and L2 Cache Controller are not supported.

19.4.5 WDOG_ENABLE eFUSE

WDOG_ENABLE eFUSE is used to enable Watchdog during Boot. By default this fuse will not be blown (0). Watchdog is configured to generate System Reset instead of Interrupt during boot. However this can be changed by application code after leaving the BootROM. Watchdog, once started, cannot be stopped or disabled, hence, it becomes application's responsibility to service watchdog before it expires.

If the fuse is 1, then the BootROM sets WDOG_WCR[WDE] to 1 with a timeout of 90 seconds. This gives 90 seconds to download, authenticate, and jump to application code. If this fails to happen, then a system reset will be issued. Note that the watchdog is serviced before jumping to application code, giving the user code 90 seconds to service the watchdog again.

In case of plugin download, watchdog is serviced before and after plugin is executed. This will mean that plugin code has to ensure that the watchdog is serviced every 90 seconds (to prevent watchdog trigger) during the period application code is downloaded to the internal memory. In case of Serial Download (USB & UART) and FLEXCAN boot modes, BOOTROM periodically services watchdog, allowing 90 seconds window for every transmission/reception of command and data

19.4.6 Exception Handling

The exception vectors located at the start of iROM (see Internal ROM / RAM Memory Map for vector table location) are used to map all the ARM exceptions.

During boot phase of primary core, all exception vectors other than reset vector point to Serial Downloader in iROM. In case of exceptions after low power exit, system will reboot by triggering Watchdog.

During the boot phase of secondary core, the exception vectors point to function that sets error status registers (see Persistent Bits), sends wakeup error interrupt and performs Wait-For-Interrupt instruction. Interrupt Service routine of primary core must reconfigure system and reset the secondary core.

19.4.7 Interrupt Handling during Boot

No special interrupt handling routines are required during the boot process. Interrupts are disabled during boot ROM execution and may be enabled later in application code.

19.4.8 Persistent Bits

Some modes of Boot ROM require registers that keep their values after warm reset. SRC General Purpose registers are used for this purpose. These bits are cleared only during Power On Reset. See the table below for persistent bits list and description.

Table 19-7. Persistent Bits

Bit Name	Bit Location	Description
PERSISTENT_ENTRY0[31:0]	SRC_GPR0[31:0]	Holds entry function for primary core for waking-up from low power mode.
PERSISTENT_ARG0[31:0]	SRC_GPR1[31:0]	Holds argument of entry function for primary core for waking-up from low power mode.
PERSISTENT_ENTRY1[31:0]	SRC_GPR2[31:0]	Holds entry function for secondary core.
PERSISTENT_ARG1[31:0]	SRC_GPR3[31:0]	Holds argument of entry function for secondary core.
PERSIST_SECONDARY_BOOT	SRC_ROM4[30]	This bit identifies which image must be used - primary and secondary. Used only for boot modes that support redundant boot.
PERSIST_BLOCK_REWRITE	SRC_ROM4[29]	Not used
PERSIST_WDOG_BOOT	SRC_ROM4[28]	This bit is set for enabling SBMR shadow register during Watchdog Reset Boot Mode. See Watchdog Reset Boot Mode for more details.
CPU1_ERROR_STATUS	SRC_ROM4[25]	Secondary core error status bit
PERSIST_SBMR_SHADOW	SRC_ROM3[31:0]	These bits are used as shadow SBMR registers during Watchdog Reset Boot Mode. See Watchdog Reset Boot Mode for more details.

19.5 Boot Devices (Internal Boot)

This device supports the following boot Flash devices/interfaces:

- Quad SPI Flash Memory
- NOR Flash with FlexBus, located on CS0 or CS1, 8, 16 or 32 bit multiplexed or non-multiplexed bus widths. (32bit non-multiplexed is not supported)
- Nand Flash memory (MLC or SLC NAND Devices), Pages sizes ranging from 2K – 8K supported.
- SD/MMC/eSD/eMMC4.3 via ESDHC interface, supporting high capacity cards

- EEPROM boot via SPI (Serial Flash) or I2C (via SPI and I2C blocks respectively).
- FlexCAN Boot.

The selection of Boot Device type is controlled by BOOT_CFG1[7:4] eFUSEs.

See the table below for more details:

Table 19-8. BOOT Device Selection

BOOT_CFG1[7:4]	Boot Device
0000	QSPI Serial Flash Memory
0001	NOR Flash (FlexBus Interface)
0010	Serial-ROM (SPI or I2C)
0011	FlexCAN Boot
0100	Reserved
0101	Reserved
0110	SD/eSD
0111	MMC/eMMC
1XXX	NAND Flash

19.5.1 QuadSPI Serial Flash Memory Boot

19.5.1.1 QuadSPI eFUSE Configuration

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7:4]	OEM	Boot Device Selection	Yes	0000	0000 – Boot from QuadSPI
BOOT_CFG1[1]	OEM	QuadSPI Interface selection	Yes	0	0 – QuadSPI0 selected 1 – QuadSPI1 selected

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 19-3 for corresponding GPIO pin.

19.5.1.2 QuadSPI Serial Flash BOOT Operation

The Boot ROM will attempt to boot from QuadSPI flash if the "BOOT_CFG1[7:4]" fuses are programmed to "0000" as shown in the QuadSPI eFUSE Configuration table. The ROM will initialize the requested the QuadSPI Interface as selected in Fuse bit BOOT_CFG1[1] in the QuadSPI eFUSE Configuration. QuadSPI interface initialization

is a two step process. The ROM expects the QuadSPI configuration parameters as explained in the QuadSPI Configuration Parameters to be present in the Serial Flash memory from the starting location of serial flash to the 318 byte offset. The ROM reads these configuration parameters using the default read command configured in the LUT of the QuadSPI interface with SCLOCK operating at 18Mhz.

In the second step, ROM configures the selected QuadSPI interface with the configuration parameters read from the serial flash and starts the boot procedure. Refer to Table 19-12 for details regarding QuadSPI configuration parameters and to the QuadSPI boot flow chart for detailed boot flow chart of QuadSPI.

Both booting an XIP and non XIP image is supported from serial flash. For XIP boot, the image has to be built for QuadSPI address space and for non XIP the image can be built to execute from DDR or SRAM.

For QUAD mode boot, the Boot ROM expects the Quad Enable bit inside the QSPI Flash to be already set before booting starts. Therefore, the QUAD enable bit must be set in the non-volatile register of the flash at the time of programming.

Note

In Case of Cortex-M4 Boot, with QSPI0, the application image entry pointer in IVT must point to QSPI0 memory region (0x2000_0000-0x2FFF_FFFF) instead of Code Alias Region (0x1000_0000-0x17ff_ffff). Application can jump to alias region for execution internally. In case of Low Power Standby Exit (for Both Cortex-A5 and Cortex-M4), Application must program STBY Exit routine in iRAM. QSPI clocks will be disabled.

19.5.1.3 IOMUX Configuration for QuadSPI

The table below shows QuadSPI IOMUX pin configuration.

Table 19-9. QuadSPI Pin Configuration

Signal	Pad Name	
	QSPI0	QSPI1
A_SCK	QSPI0_A_SCK	TRACED[3]
A_CS0	QSPI0_A_CS0	FTM0CH0
A_CS1	FTM0CH6	FTM0CH1
A_DATA[3]	QSPI0_A_DATA[3]	FTM0CH2
A_DATA[2]	QSPI0_A_DATA[2]	FTM0CH3

Table continues on the next page...

Table 19-9. QuadSPI Pin Configuration (continued)

Signal	Pad Name	
	QSPI0	QSPI1
A_DATA[1]	QSPI0_A_DATA[1]	FTM0CH4
A_DATA[0]	QSPI0_A_DATA[0]	FTM0CH5
A_DQS	QSPI0_A_DQS	-
B_SCK	QSPI0_B_SCK	-
B_CS0	QSPI0_B_CS0	-
B_CS1	FTM0CH7	-
B_DATA[3]	QSPI0_B_DATA[3]	-
B_DATA[2]	QSPI0_B_DATA[2]	-
B_DATA[1]	QSPI0_B_DATA[1]	-
B_DATA[0]	QSPI0_B_DATA[0]	-
B_DQS	QSPI0_B_DQS	-

19.5.1.4 CCM Settings in various modes

Table 19-10. QSPI0 CCM Settings

QSPI0				
Registers	**INIT**	**CLK 60**	**CLK 74**	**CLK 99**
Clock Source	PLL3_PFD4	PLL3_MAIN	PLL3_PFD4	PLL3_PFD4
CCSR	0x8004_0824	0x0004_0824	0x8004_0824	0x8004_0824
CSCMR1	0x0040_0000	0x0000_0000	0x0040_0000	0x0040_0000
CSCDR3	0x0000_001F	0x0000_001D	0x0000_001C	0x0000_0012

Table 19-11. QSPI1 CCM Settings

QSPI1				
Registers	**INIT**	**CLK 60**	**CLK 74**	**CLK 99**
Clock Source	PLL3_PFD4	PLL3_MAIN	PLL3_PFD4	PLL3_PFD4
CCSR	0x8004_0824	0x0004_0824	0x8004_0824	0x8004_0824
CSCMR1	0x0100_0000	0x0000_0000	0x0000_0000	0x0000_0000
CSCDR3	0x0000_1F00	0x0000_1D00	0x0000_1C00	0x0000_1200

19.5.1.5 QuadSPI Configuration Parameters

Table below lists various QuadSPI Configuration Parameters.

Table 19-12. QuadSPI Configuration Parameters

<table>
<tr><th>Name</th><th>Offset</th><th>Size in Bytes</th><th>Description</th></tr>
<tr><td>Reserved</td><td>0</td><td>4</td><td>Reserved to 0
NOTE: Any other value may result in undefined operation</td></tr>
<tr><td>Hold Delay</td><td>4</td><td>1</td><td>Hold Delay for QSPI[0,1] A/B<table><tr><th>Value</th><th>QSPI0 B</th><th>QSPI0 A/QSPI1 A</th></tr><tr><td>00</td><td>Disable</td><td>Disable</td></tr><tr><td>01</td><td>Disable</td><td>Enable</td></tr><tr><td>10</td><td>Enable</td><td>Disable</td></tr><tr><td>11</td><td>Enable</td><td>Enable</td></tr></table></td></tr>
<tr><td>Half Speed Phase Selection</td><td>5</td><td>1</td><td>Half Speed Phase Selection
0 Select sampling at non-inverted clock
1 Select sampling at inverted clock</td></tr>
<tr><td>Half Speed Delay Selection</td><td>6</td><td>1</td><td>Half Speed Delay Selection
0 One Clock cycle delay
1 Two Clock cycle delay</td></tr>
<tr><td>Reserved1</td><td>7</td><td>1</td><td>This word is reserved</td></tr>
<tr><td>Reserved2</td><td>8</td><td>4</td><td>This word is reserved</td></tr>
<tr><td>Reserved3</td><td>12</td><td>4</td><td>This word is reserved</td></tr>
<tr><td>Reserved4</td><td>16</td><td>4</td><td>This word is reserved</td></tr>
<tr><td>Chip Select hold time</td><td>20</td><td>4</td><td>This is chip select hold time in terms of Serial clock (For Example 1 serial clock cycle).</td></tr>
<tr><td>Chip Select setup time</td><td>24</td><td>4</td><td>Chip select setup time in terms of Serial clock (For example 1 serial clock).</td></tr>
<tr><td>Serial Flash A1 size</td><td>28</td><td>4</td><td>Serial Flash A1 size in units of Bytes</td></tr>
<tr><td>Serial Flash A2 size</td><td>32</td><td>4</td><td>Serial Flash A2 size in units of bytes</td></tr>
<tr><td>Serial Flash B1 size</td><td>36</td><td>4</td><td>Serial Flash B1 size in units of bytes</td></tr>
<tr><td>Serial Flash B2</td><td>40</td><td>4</td><td>Serial Flash B2 size in units of bytes</td></tr>
<tr><td>Serial Clock Frequency</td><td>44</td><td>4</td><td>This is serial clock frequency select parameter.<table><tr><th>Value</th><th>Clock</th></tr><tr><td>00</td><td>18 MHz</td></tr><tr><td>01</td><td>60 MHz</td></tr><tr><td>02</td><td>74 MHz</td></tr><tr><td>03</td><td>99Mhz (only SDR mode)</td></tr></table></td></tr>
<tr><td>Reserved5</td><td>48</td><td>4</td><td>This field is reserved</td></tr>
</table>

Table continues on the next page...

Table 19-12. QuadSPI Configuration Parameters (continued)

Name	Offset	Size in Bytes	Description
Mode of operation of serial Flash	52	1	This field describes the mode of operation of Serial flash Value / Mode: 01 – Single 02 – Dual 04 – Quad
Serial Flash Port B Selection	53	1	Port A is always available. This field informs the device ROM the availability of Port B. 0 – Port B is not used 1 – Port B is used
Dual Data Rate mode enable	54	1	This field enables the device ROM to enable DDR mode in QSPI. 0 – DDR mode is disabled 1 – DDR mode is enabled
Data Strobe Signal enable in Serial Flash	55	1	This field enables Data Strobe signal in Serial Flash which supports it. 0 – Disable DQS 1 – Enable DQS
Parallel Mode enable	56	1	This field enables the device ROM to configure the QSPI interface in parallel mode. Data will be read from serial Flash in parallel mode. Refer to QSP chapter for detail. 0 – Disable Parallel mode in QSPI 1 – Enable Parallel Mode in QSPI
CS1 on Port A	57	1	This field helps ROM to enable CS1 on port A 0 – Disable CS1 on Port A 1 – Enable CS1 on Port A
CS1 on Port B	58	1	This field helps ROM to enable CS1 on port B 0 – Disable CS1 on Port B 1 – Enable CS1 on Port B
Full Speed Phase Selection	59	1	Select the edge of the sampling clock valid for full speed commands: 0: Select sampling at non-inverted clock 1: Select sampling at inverted clock This bit is also used to shift the dqs_enable when DQS mode is selected
Full Speed Delay Selection	60	1	Select the delay w.r.t. the reference edge for the sample point valid for full speed commands: 0: One clock cycle delay 1: Two clock cycles delay This bit is also used to shift the dqs_enable when DQS mode is selected
DDR Sampling Point	61	1	Select the sampling point for incoming data when serial flash is in DDR mode. **NOTE:** Valid Values are (b000-b111)

Table continues on the next page...

Table 19-12. QuadSPI Configuration Parameters (continued)

Name	Offset	Size in Bytes	Description
LUT program sequence	62	256	256 Bytes of Look up table program sequence. ROM programs the LUT of QuadSPI with this parameter supplied. It assumes that the optimize read command sequence which will be used to read data from Serial flash and fill the AHB buffer is programmed at index 0.

The table below shows the available frequency in System Boot for booting through QuadsSPI0 with multiple configuration (SDR, DDR modes):

Table 19-13. QuadSPI SCK frequency limitation in Boot ROM

	QuadSPI0 in DDR Mode	QuadSPI0 in SDR Mode
SCK frequency options in MHz	18	18, 60, 74

19.5.1.6 QuadSPI boot flow chart

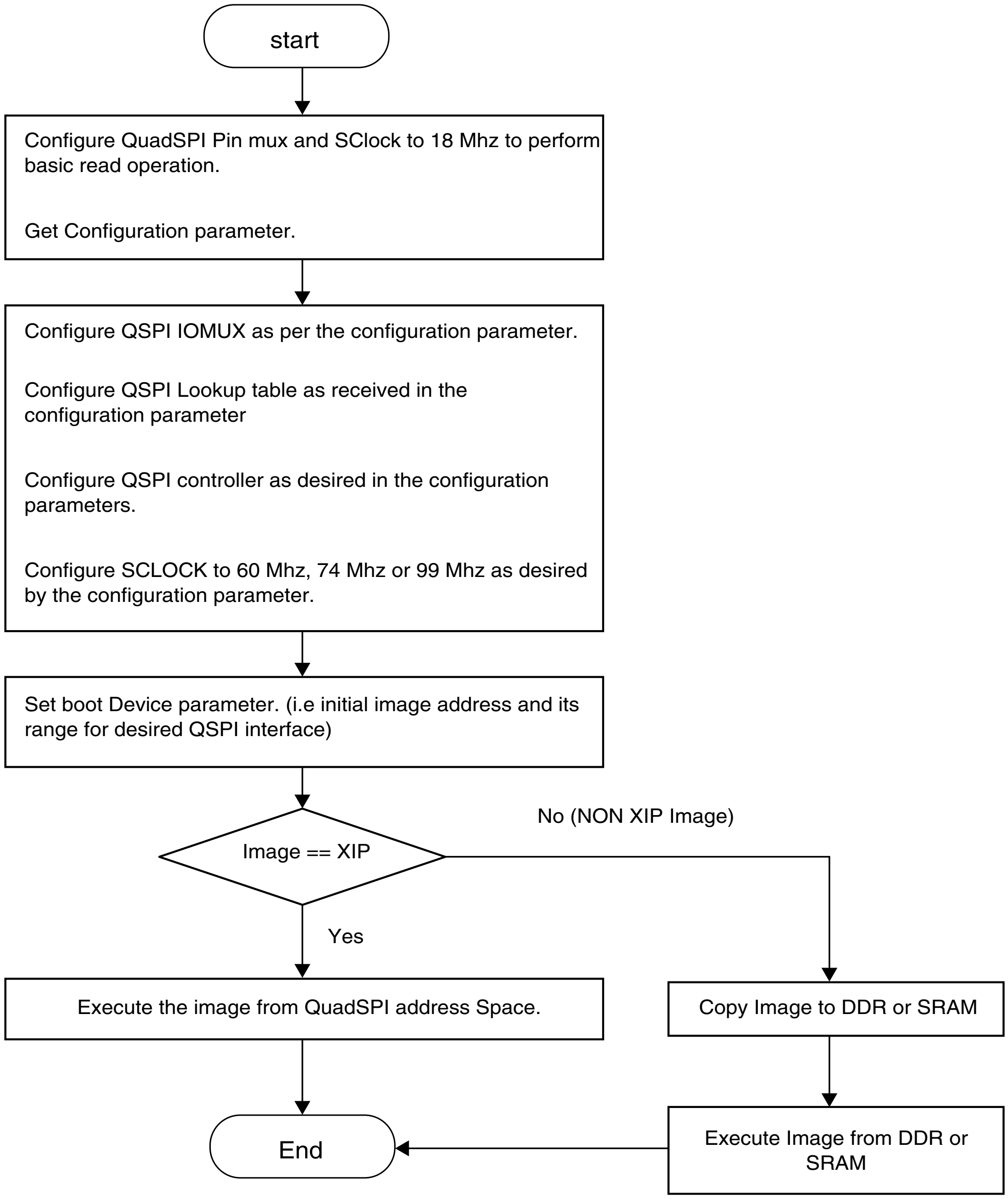


Figure 19-3. QuadSPI boot flow chart

NOTE

If flash is configured for "High performance mode (where command is generated only once)" in LUT program sequence. then external reset should be routed to flash reset to allow rebooting in case of any device reset other than Power On Reset. Also this high performance mode must be exited by application before any Low power mode entry where the device is supposed to reboot from QSPI flash on Low power mode

exit. In general, any preserved configuration in external flash will not be understood by device after reset.

19.5.2 NOR Flash Boot using FlexBus Interface

19.5.2.1 NOR Flash eFUSE Configuration

The FlexBus module works in the asynchronous mode, and supports muxed, Address/ Data, or non-muxed schemes based on fuse settings:

Table 19-14. FlexBus BOOT eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7:4]	OEM	Boot Device Selection	Yes	0001	0010 – Boot from FlexBus
BOOT_CFG1[3]	OEM	Chip Select	Yes	0	0 – Boot from FB_CS0 1 – Boot from FB_CS1
BOOT_CFG1[2:1]	OEM	Port Size Select	Yes	00	1X – 16 bit 01 – 8 bit 00 – 32 bit
BOOT_CFG1[0]	OEM	Multiplexed/Non-Multiplexed configuration	Yes	0	0 – Non Muxed Mode 1 – Muxed Mode
BOOT_CFG2[4:5]	OEM	Address Select	Yes	00	00 – Assert FB_CSn on the first rising clock edge after the address is asserted. 01 – Assert FB_CSn on the second rising clock edge after the address is asserted. 10 – Assert FB_CSn on the third rising clock edge after the address is asserted. 11 – Assert FB_CSn on the fourth rising clock edge after the address is asserted.
BOOT_CFG2[3]	OEM	Extended Latch Enable	Yes	0	0 – FB_TS/ALE assert for one cycle 1 – FB_TS/ALE asserted until the first positive edge after FB_CS0
BOOT_CFG2[1]	OEM	Byte Enable	Yes	0	0 – No Byte enable asserted for Read 1 – Byte enable asserted for Read
BOOT_CFG2[0]	OEM	Transfer Acknowledge	Yes	0	0 – Internal Transfer ACK 1 – External Transfer ACK

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 19-3 for corresponding GPIO pin.

19.5.2.2 NOR Flash Boot Operation

Booting from the NOR Flash is supported via FlexBus interface. The FlexBus interface clock is fixed to **33MHz**. The device ROM reads NOR Flash Configuration Block (see NOR Flash Configuration Block) directly from Flash. Thereafter configures FlexBus Controller parameters for optimized operation. The FlexBus IO interface clock can be increased by programming the Clock divider in FlexBus Configuration Block. Subsequently, Image Vector Table and Boot Data structures are read to determine if the image can be executed directly from FlexBus address space or should be copied to other memory. The start field of Boot Data Structure specifies the final location of the image

(see Image Vector Table and Boot Data).

NOTE

In Case of Cortex-M4 Boot, the application image entry pointer in IVT must point to the FlexBus memory region (0x3000_0000-0x3FFF_FFFF) instead of the Code Alias Region (0x1800_0000-0x1eff_ffff). Application can jump to alias region for execution internally.

In case of Low Power Standby Exit (for Both Cortex-A5 & Cortex-M4), Application must program STBY Exit routine in iRAM. FlexBus clocks will be disabled.

19.5.2.3 NOR Flash Configuration Block

NOR Flash Configuration Block is present at an offset of **0x00**, from the start of Flash memory. The block contains information for programming the FlexBus Controller.

Details of the NOR Flash Configuration Block:

Name	Start Byte	Size in Bytes	Description
Finger Print	0	4	32 bit word with a value of "0x11223344"
Flash Size	4	4	Size of NOR Flash connected
Write Protect	8	1	Flash Write Protect 0 – Disable 1 – Enable

Table continues on the next page...

Name	Start Byte	Size in Bytes	Description
Secondary Wait State Enable Flag	9	1	Flash Secondary Wait State 0 – Disable 1 – Enable
Secondary Wait State Value	10	1	Secondary Wait state Value. Used only when Secondary Wait State Enable Flag is Set.
Address Setup	11	1	Controls the Assertion of chip-select with respect to assertion of a valid address and attributes. The address and attributes are considered valid at the same time FB_ALE asserts. 00 Assert FB_CS*n* on first rising clock edge after address is asserted. (Default FB_CS*n*) 01 Assert FB_CS*n* on second rising clock edge after address is asserted. 10 Assert FB_CS*n* on third rising clock edge after address is asserted. 11 Assert FB_CS*n* on fourth rising clock edge after address is asserted. (Default FB_CS0)
Read Address Hold	12	1	This field controls the address and attribute hold time after the termination during a read cycle that hits in the chip-select address space. The hold time applies only at the end of a transfer. Therefore, during a burst transfer or a transfer to a port size smaller than the transfer size, the hold time is only added after the last bus cycle. The number of cycles the address and attributes are held after FB_CS*n* negation depends on the value of CSCR*n*[AA] as shown below.
Write Address Hold	13	1	Write address hold or deselect. This field controls the address, data, and attribute hold time after the termination of a write cycle that hits in the chip-select address space. The hold time applies only at the end of a transfer. Therefore, during a burst transfer or a transfer to a port size smaller than the transfer size, the hold time is only added after the last bus cycle. 00 Hold address and attributes one cycle after FB_CSn negates on writes. (Default FB_CSn) 01 Hold address and attributes two cycles after FB_CSn negates on writes. 10 Hold address and attributes three cycles after FB_CSn negates on writes. 11 Hold address and attributes four cycles after FB_CSn negates on writes. (Default FB_CS0)
Wait State	14	1	The number of wait states inserted after FB_CSn asserts and before an internal transfer acknowledge is generated (WS = 0 inserts zero wait states, WS = 0x3F inserts 63 wait states). If AA is reserved, FB_TA must be asserted by the external system regardless of the number of generated wait states. In that case, the external transfer acknowledge ends the cycle. An external FB_TA supersedes the generation of an internal FB_TA.
Clock Divider	15	1	The value between (1 & 8). This is used to configure the Flexbus Divider in CCM. CLK_{FBUS} = Platform Bus Clock / (Clock Divider - 1)

Table continues on the next page...

Name	Start Byte	Size in Bytes	Description
Burst-read enable.	16	1	Specifies whether burst reads are used for memory associated with each FB_CSn. 0 Data exceeding the specified port size is broken into individual, port-sized, non-burst reads. For example, a longword read from an 8-bit port is broken into four 8-bit reads. 1 Enables data burst reads larger than the specified port size, including longword reads from 8- and 16-bit ports, word reads from 8-bit ports, and line reads from 8, 16-, and 32-bit ports.
Burst-write enable	17	1	Specifies whether burst writes are used for memory associated with each FB_CSn. 0 Break data larger than the specified port size into individual, port-sized, non-burst writes. For example, a longword write to an 8-bit port takes four byte writes. 1 Enables burst write of data larger than the specified port size, including longword writes to 8 and 16-bit ports, word writes to 8-bit ports, and line writes to 8-, 16-, and 32-bit ports.
Reserved	18	14	Not Used by ROM

19.5.2.4 CCM Settings in various modes

Table 19-15. FlexBUS CCM Settings

FlexBus		
Registers	**INIT**	**FINAL**
Clock Source	PLL1_PFD4	PLL1_PFD4
CACRR	0x00E0_0809	Configurable by “Clock Divider” field in NOR Flash Configuration Block

19.5.2.5 IOMUX Configuration for FlexBus

The table below shows FlexBUS IOMUX pin configuration.

Table 19-16. FlexBus Pin Configuration

Signal	Pad Name
FB_AD[31]	FB_AD[31]
FB_AD[30]	FB_AD[30]
FB_AD[29]	FB_AD[29]
FB_AD[28]	FB_AD[28]
FB_AD[27]	FB_AD[27]
FB_AD[26]	FB_AD[26]

Table continues on the next page...

Table 19-16. FlexBus Pin Configuration (continued)

Signal	Pad Name
FB_AD[25]	FB_AD[25]
FB_AD[24]	FB_AD[24]
FB_AD[23]	FB_AD[23]
FB_AD[22]	FB_AD[22]
FB_AD[21]	FB_AD[21]
FB_AD[20]	FB_AD[20]
FB_AD[19]	FB_AD[19]
FB_AD[18]	FB_AD[18]
FB_AD[17]	FB_AD[17]
FB_AD[16]	FB_AD[16]
FB_AD[15]	QSPI0_A_SCK
FB_AD[14]	QSPI0_A_CS0
FB_AD[13]	QSPI0_A_DATA[3]
FB_AD[12]	QSPI0_A_DATA[2]
FB_AD[11]	QSPI0_A_DATA[1]
FB_AD[10]	QSPI0_A_DATA[0]
FB_AD[9]	QSPI0_A_DQS
FB_AD[8]	QSPI0_B_SCK
FB_AD[7]	QSPI0_B_CS0
FB_AD[6]	QSPI0_B_DATA[3]
FB_AD[5]	QSPI0_B_DATA[2]
FB_AD[4]	QSPI0_B_DATA[1]
FB_AD[3]	QSPI0_B_DATA[0]
FB_AD[2]	QSPI0_B_DQS
FB_AD[1]	SCI0_RTS
FB_AD[0]	SCI0_CTS
FB_ALE	SAI0_TX_BCLK
FB_CS1_b	SAI0_RX_DATA
FB_CS0_b	SAI0_TX_DATA
FB_OE_b	SAI0_RX_SYNC
FB_R/W_b	SAI0_TX_SYNC
FB_TA_b	SAI1_TX_BCLK
FB_CLKOUT	FTM0CH6
FB_BE3_b	SAI1_RX_BCLK
FB_BE2_b	SAI1_RX_DATA
FB_BE1_b	SAI1_TX_DATA
FB_BE0_b	SAI1_RX_SYNC

19.5.3 Serial ROM Boot using SPI/I2C Interface

The device supports boot from serial memory devices such as EEPROM and Serial Flash, using SPI (SPI 1, SPI 2, SPI 3, SPI 4), and I2C Controller (I2C 1, I2C 2, I2C 3, and I2C 4) interfaces.

19.5.3.1 Serial ROM eFUSE Configuration

The boot ROM code determines the type of device using the following parameters, either provided by eFUSE settings or sampled on the I/O pins, during boot (See the table below for details):

Table 19-17. Serial ROM BOOT eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7:4]	OEM	Boot Device Selection	Yes	0000	0010 – Boot from Serial ROM
BOOT_CFG4[5:4]	OEM	CS Select (SPI Only)	Yes	00	00 – CS#0 01 – CS#1 10 – CS#2 11 – CS#3
BOOT_CFG4[3]	OEM	SPI Addressing (SPI Only)	Yes	0	0 – 2 bytes (16 bit) 1 – 3 bytes (24 bit)
BOOT_CFG4[2:0]	OEM	Port Select	Yes	000	000 – SPI0 001 – SPI1 010 – SPI2 011 – SPI3 100 – IIC0 101 – IIC1 110 – IIC2 111 – IIC3

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 19-3 for corresponding GPIO pin.

The I2C-0/I2C-1/I2C-2/I2C-3 block can be used as boot device using I2C interface, for serial ROM boot. The I2C interface is configured to operate at 343.75 Kbps. The boot ROM will copy 4Kbyte of data from Serial ROM device to internal RAM. After checking the Image Vector Table header value (0xD1) from Program Image, the ROM code performs a DCD check. After successful DCD extraction, the Rom code extracts from Boot Data Structure the destination pointer and length of image to be copied to RAM device from where code execution occurs.

Note

The Initial 4K of Program Image must contain the IVT, DCD and the Boot Data structures.

19.5.3.2 SPI Boot

The Serial Peripheral Interface (SPI) interface is configured in Master mode and the EEPROM device is connected to SPI interface as slave. The boot ROM code copies 4 Kbyte data from EEPROM device to the internal RAM. If DCD verification is successful, the ROM code copies the initial 4 Kbyte data as well as the rest of image directly to application destination extracted from application image. The SPI can read data from EEPROM using 2 or 3 byte addressing and its burst length is 32 bytes.The Clock Polarity(CPOL) and Clock Phase(CPHA) bits setting of the SPI block SPIx_CTARn register are set as below during boot –

SPIx_CTARn[CPOL] = 0

SPIx_CTARn[CPHA] = 0

Note

The Serial ROM Chip Select Number is determined by BOOT_CFG4 [5:4] (Chip Select) fuse.

When using the SPI as boot device, the Vybrid supports booting from both Serial EEPROM and Serial Flash devices. The boot code determines which device is being used by reading the appropriate eFUSE/I/O values at boot (See Table 19-17 for details).

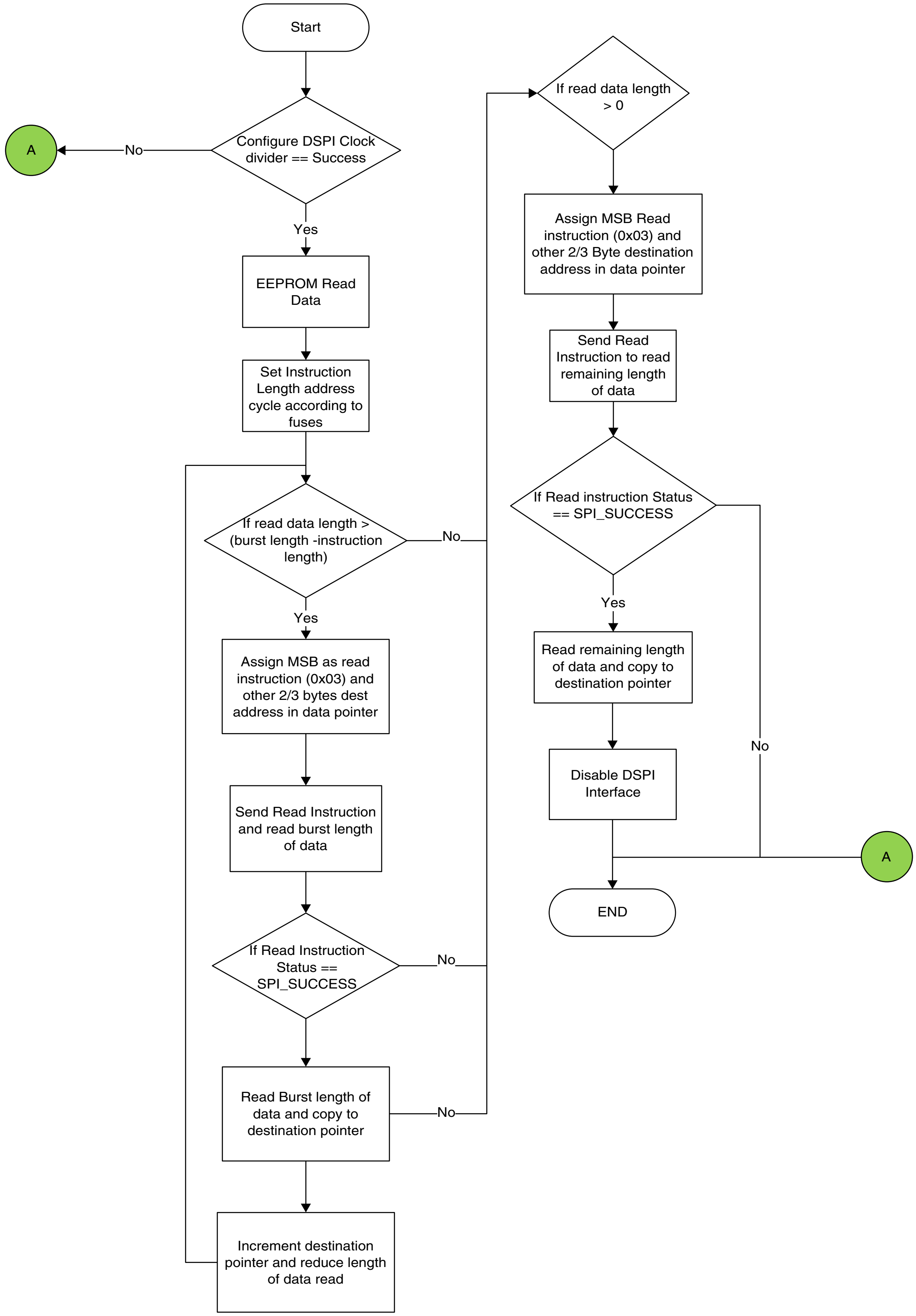


Figure 19-4. SPI Boot Flow Diagram

19.5.3.3 IOMUX Configuration for SPI

The table below shows SPI IOMUX pin configuration.

Table 19-18. SPI Pin Configuration

Signal	Pad Name			
	SPI0	SPI1	SPI2	SPI3
CS3	SAI1_RX_DATA	QSPI0_A_DATA[3]	-	-
CS2	SAI1_TX_DATA	QSPI0_A_DATA[2]	-	-
CS1	SPI0_PCS1	QSPI0_A_DATA[1]	FB_AD[31]	QSPI0_B_DATA[3]
CS0	SPI0_PCS0	QSPI0_A_DATA[0]	FB_AD[30]	QSPI0_B_DATA[2]
SIN	SPI0_SIN	QSPI0_A_DQS	FB_AD[29]	QSPI0_B_DATA[1]
SOUT	SPI0_SOUT	QSPI0_B_SCK	FB_AD[28]	QSPI0_B_DATA[0]
SCK	SPI0_SCK	QSPI0_B_CS0	FB_AD[27]	QSPI0_B_DQS

19.5.3.4 I2C Boot

The boot flow when booting from an I2C device is shown in Figure 19-5. The boot ROM code reads the fuses BOOT_CFG4[2:0] (Boot Device Selection) and BOOT_CFG1 [7:4] (Port select) to detect EEPROM device type. The ROM program copies 4K data from the EEPROM device to internal RAM. The boot ROM code next copies the initial 4Kbyte of data as well as rest of image directly to application destination extracted from application image.

The device uses the Device Select Code/Device Address in Table 19-19 to boot from an EEPROM.

Table 19-19. EEPROM via I2C Device Select Code

Bits	Device Type Identifier				Chip Enable Address[1]			R/W
	7	6	5	4	3	2	1	0
Device Select Code	1	0	1	0	0	0	0	R/W

1. These address bits, should be configured at the memory device, to match this '000' value.

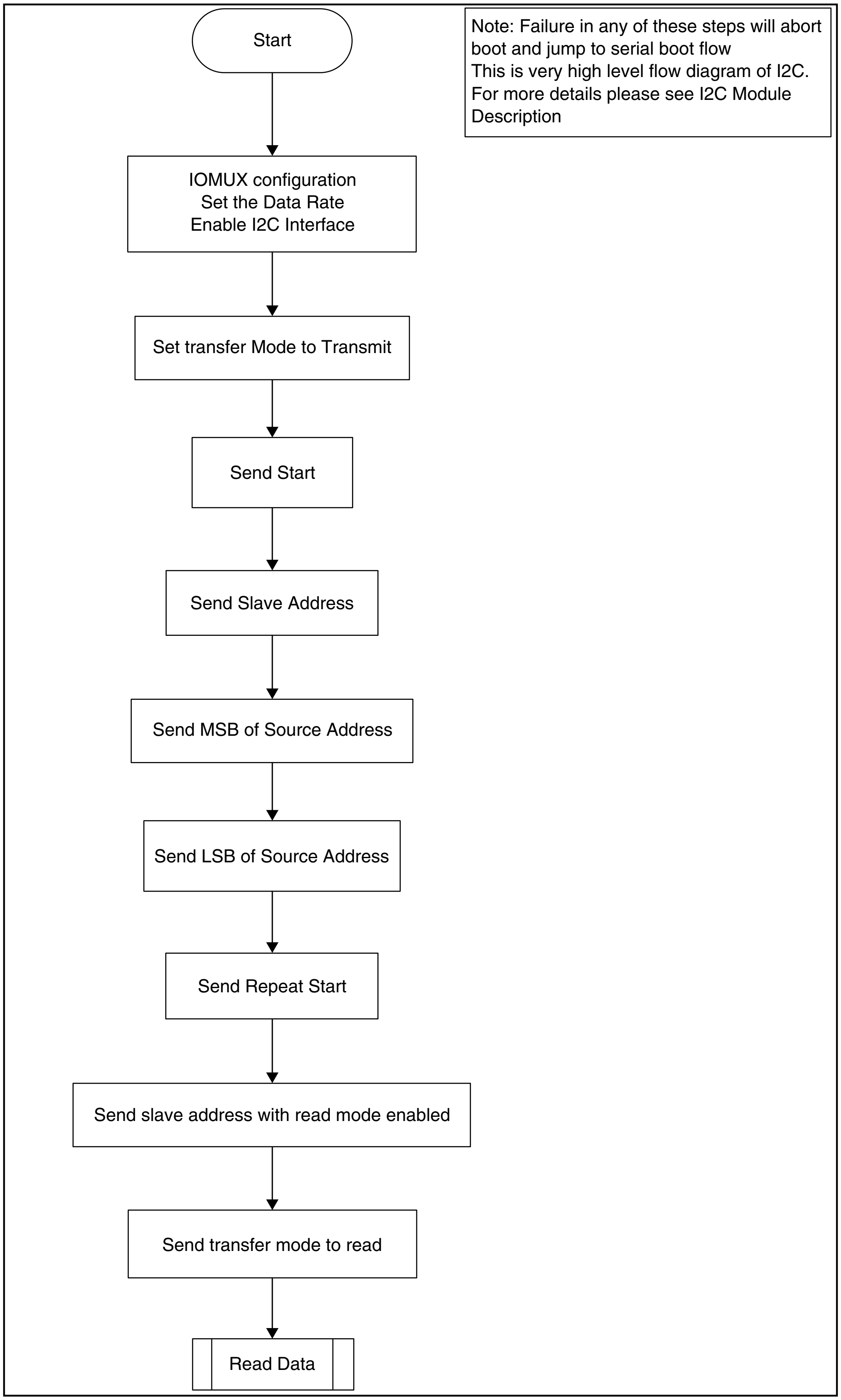


Figure 19-5. I2C Boot Flow Diagram

19.5.3.5 IOMUX Configuration for I2C

The table below shows I2C IOMUX pin configuration.

Table 19-20. I2C Pin Configuration

Signal	Pad Name			
	I2C0	I2C1	I2C2	I2C3
SCL	TRACECK	TRACED[1]	TRACED[6]	TRACED[14]
SDA	TRACED[0]	TRACED[2]	TRACED[7]	TRACED[15]

19.5.4 FlexCAN Boot

This device supports boot from CAN interfaces, either from FlexCAN0 or FlexCAN1 ports.

19.5.4.1 FlexCAN eFUSE Configuration

BOOTROM code determines the type of device using the following parameters; either provided by eFUSE settings or sampled on the I/O pins, during boot (See the table below for details):

Table 19-21. BOOT Device Selection

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7:4]	OEM	Boot Device Selection	Yes	0000	0011 – Boot From FlexCAN
BOOT_CFG1[0]	OEM	FlexCAN Port Selection	Yes	0	0 – Boot From FlexCAN0 1 – Boot From FlexCAN1

1. Setting can be overridden by GPIO settings when GPIO_FUSE_SEL fuse is intact. See Table 19-3 for corresponding GPIO pin.

19.5.4.2 FlexCAN Configuration Parameters

Either FlexCAN0 or FlexCAN1 ports can be used to download the image from an external CAN host. The FlexCAN interface is configured to operate at 1 Mbps.

Table 19-22. Bit timing parameters in FlexCAN

FlexCAN Parameter	Value (in Time Quanta)	CAN2.0 Parameter	Value (in Time Quanta)
SYNC_SEG	1	SYNC_SEG = FlexCAN SYNC_SEG	1
PROP_SEG	3	PROP_SEG = FlexCAN PROP_SEG+1	4
PSEG1	2	PHASE_SEG1 = FlexCXAN PSEG1+1	3
PSEG2	3	PHASE_SEG2 = FlexCAN PSEG2+1	4

19.5.4.3 FlexCAN Boot Operation

FlexCAN operates as a device, which is to be connected to an external CAN Host. The Host transfers the application image as per Serial Download Protocol. BOOTROM configures device ID as 0x04 and responds to Host with ID 0x14.

Before downloading the image, Host and device have to be associated. During this phase, the device waits for a pattern (0x67898967) from the host. After the pattern is received, it then responds with same pattern back to the host. This completes the ASSOCIATION Phase. BOOTROM waits for a period of 60 seconds for the ASSOCATION Phase before switching to Serial Download Mode (using UART/USB).

After Association phase is completed, the Host can download the application image using the Serial Download protocol. The Serial Download Protocol has the capability of DCD_WRITE before the image is downloaded to destination memory. Once the image is downloaded, CAN host must issue a JUMP command to the Vybrid CAN device so that BOOTROM can authenticate and the jump to the entry point of the downloaded image.

A data flow control mechanism is required between the host and device for downloading image over FlexCAN interface. A software flow control mechanism has been developed below SDP protocol layer as shown in the following figure:

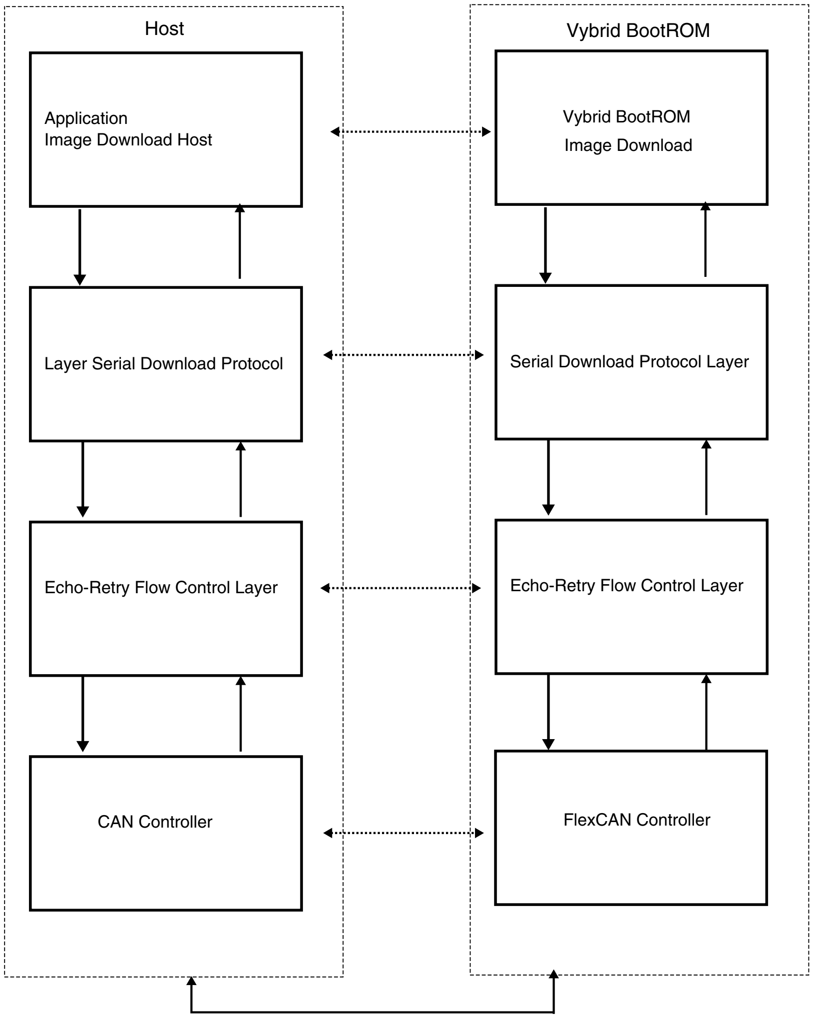


Figure 19-6. FlexCAN Boot Protocol

From Bottom up, the CAN boot architecture is similar to OSI layer architecture consisting of Physical, Data Link/flow control, Transport and Application layers only.

On top of FlexCAN controller, BOOTROM implements an echo-retry flow control mechanism in software. This mechanism has a concept of echoing back the data by the receiver which has been sent by a sender. In this architecture, a sender is an entity who has some data to send (as an originator) as per the upper layer protocol, which is SDP here. For example, SDP commands are always sent by Host, so the host will be called a sender, whereas Vybrid device will be called a receiver. Again, as the SDP responses are always sent back by Vybrid device, so the device will be a sender in this case, and host being a receiver, Echo back mechanism assures the sender doesn’t send the next data to the receiver as it has to wait to receive the echo data of the last data sent to the intended receiver, which is Vybrid device here.

Only Echo back mechanism is not enough for stable flow control, as SDP protocol has data flowing from both Host (command, data) and Devices (response, data). So in cases (especially when there is a switch between the sender and receiver functions between the host and device, e.g., sending command to receiving response or otherwise), either entity may miss a data. So a Retry mechanism is implemented to take care of this scenario. To be able to download the image properly, the host application has to implement the layered application as described above.

NOTE

ECC detection and correction operation of FlexCAN internal RAM Memory is enabled by default. Boot Code thus initializes FlexCAN internal RAM area only if FlexCAN is selected for boot operation. During boot operation, if any un -correctable Errors detected, FlexCAN port is put into Freeze mode and the boot through FlexCAN is failed (and then may jump to recovery interface or serial downloader, as the case may be).

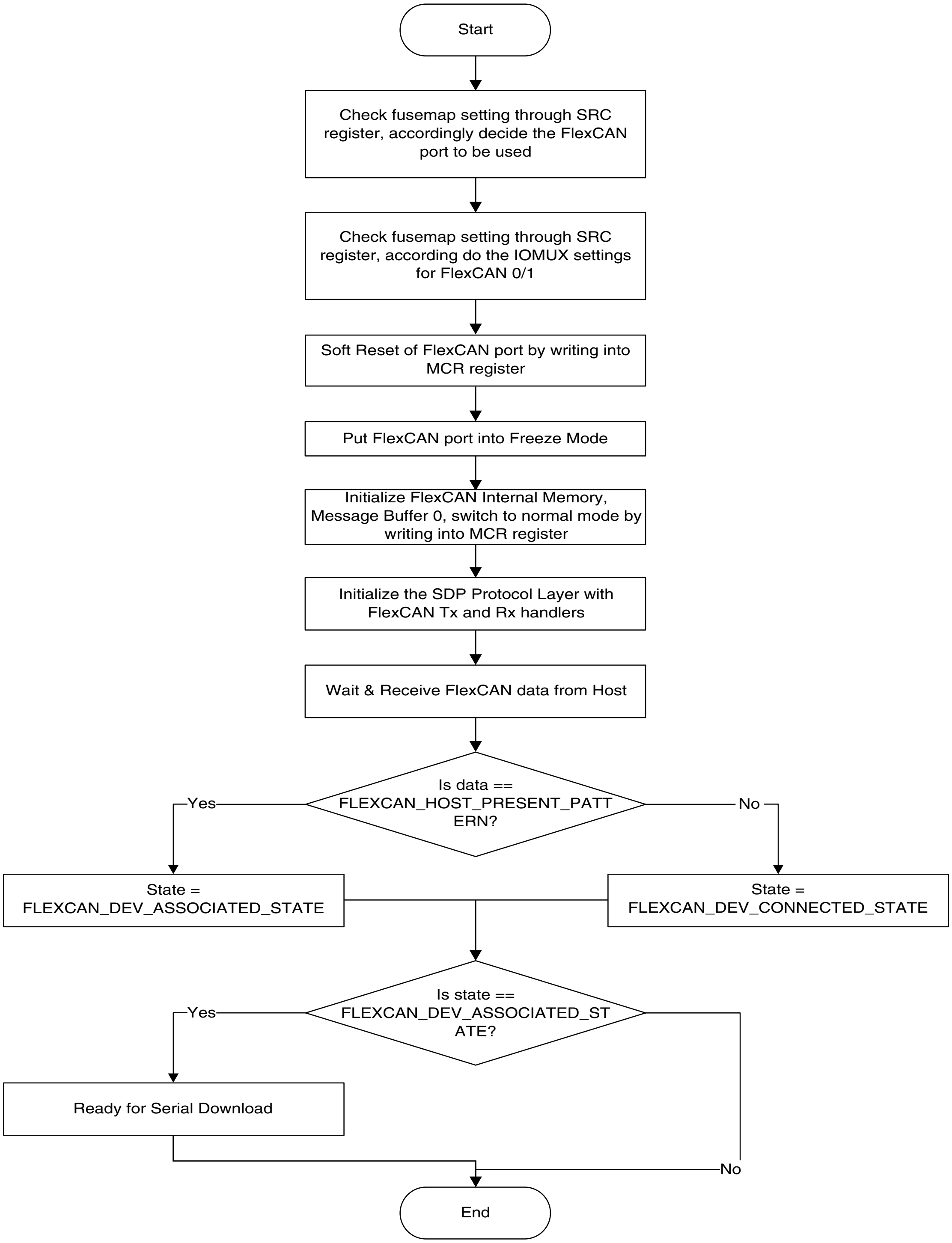


Figure 19-7. FlexCAN Flow Chart

19.5.4.4 IOMUX Configuration for FlexCAN

The table below shows FlexCAN IOMUX pin configuration.

Table 19-23. FlexCAN Pin Configuration

Signal	Pad Name	
	FlexCAN0	FlexCAN1
RX	CAN0_RX	CAN1_RX
TX	CAN0_TX	CAN1_TX

19.5.5 SD/MMC Boot

The device ROM supports booting from MMC/eMMC and SD/eSD compliant devices.

19.5.5.1 SD/MMC eFUSE Configuration

SD/MMC/eSD/eMMC boot can be performed using either ESDHC-0, ESDHC-1 ports, based on setting of the BOOT_CFG2 [3] (Port Select) fuse or its associated GPIO input value at boot. All eSDHC ports support eMMC4.3 and fast boot. See the table below for details.

Table 19-24. ESDHC BOOT eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7:5]	OEM	Boot Device Selection	Yes	000	011 – BOOT from eSDHC interface
BOOT_CFG1[4]	OEM	SD/MMC Selection	Yes	0	0 – SD/eSD 1 – MMC/eMMC
BOOT_CFG1[3]	OEM	Fast Boot Enable	Yes	0	0 – Fast Boot Enable 1 –Fast Boot Disable
BOOT_CFG1[1]	OEM	SD /MMC Speed	Yes	0	1 – Normal 0 – High
BOOT_CFG2[7:5]	OEM	Bus Width/SD Calibration Step	Yes	000	SD/eSD (BOOT_CFG1[5]=0) Bus Width xx0 - 1-bit xx1 - 4-bit MMC/eMMC (BOOT_CFG1[5]=1) x00 - 1-bit x01 - 4-bit x10 - 8-bit Else - reserved.

Table continues on the next page...

Table 19-24. ESDHC BOOT eFUSE Descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG2[3]	OEM	Port Select	Yes	0	0 – ESDHC0 1 – ESDHC1
BOOT_CFG2[1]	OEM	Fast Boot Acknowledge Disable (eMMC Only)	Yes	0	0 - Boot Acknowledge Enabled. 1 - Boot Acknowledge Disabled.
BOOT_CFG2[0]	OEM	Override Pad Settings	Yes	0	0 - Use default values 1 - Use PAD_SETTINGS values

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 19-3 for corresponding GPIO pin.

Boot code supports following standards.

- MMCv4.3 or less
- eMMCv4.3 or less
- SDv2.0 or less
- eSDv2.10 rev-0.9, with or without FAST_BOOT.

MMC/SD/eSD/eMMC can be connected to any of ESDHC-0, 1 block and can be booted by copying 4Kbyte of data from MMC/SD/eSD/eMMC device to internal RAM. After checking the Image Vector Table header value (0xD1) from Program Image, the ROM code performs a DCD check. After successful DCD extraction, the ROM code extracts from Boot Data Structure the destination pointer and length of image to be copied to RAM device from where code execution occurs.

Note

The Initial 4Kbyte of Program Image must contain the IVT, DCD and the Boot Data structures.

Table 19-25. SD/MMC Frequencies

	SD	MMC
Identification (KHz)	400 KHz	
Normal Speed Mode (MHz)	22 MHz	18.62 MHz
High Speed Mode (MHz)	44 MHz	42.56 MHz

Note

BOOTROM code reads application image length and application destination pointer from image.

19.5.5.2 MMC and eMMC Boot

The following table provides MMC and eMMC boot details.

Table 19-26. MMC and eMMC Boot Details

Normal Boot Mode	During initialization (normal boot mode) the MMC frequency is set to 400 KHz. When the MMC card enters the identification portion of the initialization, voltage validation is performed and the ROM boot code checks high voltage settings and card capacity. The ROM boot code supports both high capacity and low capacity MMC/eMMC cards. After initialization phase is complete, the ROM boot code switches to a higher frequency (18.62 MHz in Normal boot mode or 42.56 MHz in High Speed mode). eMMC is also interfaced via eSDHC and follows the same flow as MMC. The boot partition can be selected for an MMC4.x card after the card initialization is complete. The ROM code reads the BOOT_PARTITION_ENABLE field in the Ext_CSD [179] to get the boot partition to be set. If there is no boot partition mentioned in BOOT_PARTITION_ENABLE field or the user partition has been mentioned, ROM boots from the user partition.
eMMC4.3 Device Supporting Special Boot Mode	If using an eMMC4.3 device supporting special boot mode, it can be initiated by pulling the CMD line low. If BOOT ACK is enabled, the eMMC4.3 device sends the BOOT ACK via DATA lines and ROM can read the BOOT ACK [S010E] to identify the eMMC4.3 device. If BOOT ACK is enabled ROM waits 50 ms to get the BOOT ACK and if BOOT ACK is received by ROM. If BOOT ACK is disabled ROM waits 1 second for data. If BOOT ACK or data was received then eMMC4.3 is booted in "Boot mode", otherwise eMMC4.3 boots as a normal MMC card from the selected boot partition. This boot mode can be selected by BOOT_CFG1 [3] (Fast Boot) fuse. BOOT ACK is selected by BOOT_CFG2 [1].

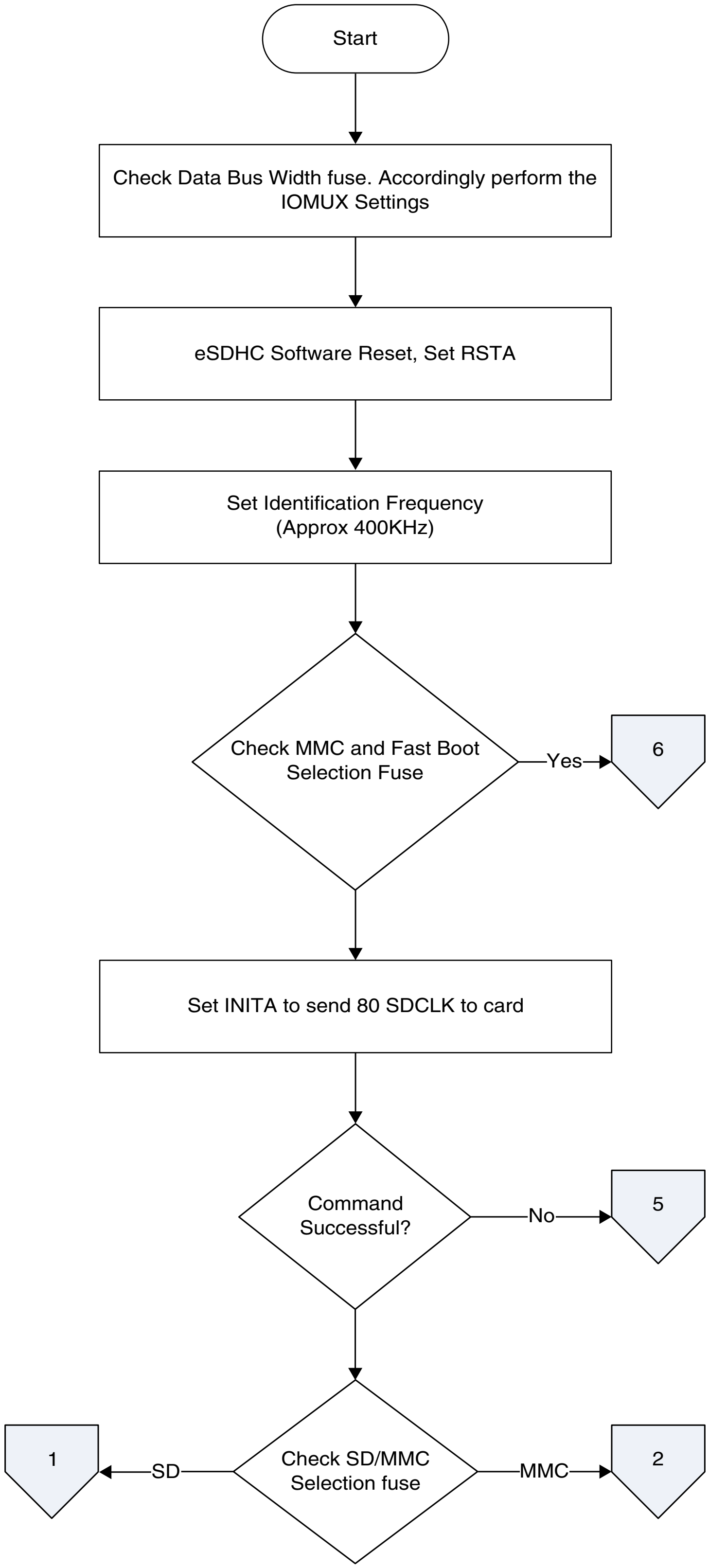


Figure 19-8. Expansion Device Boot Flow (1/7)

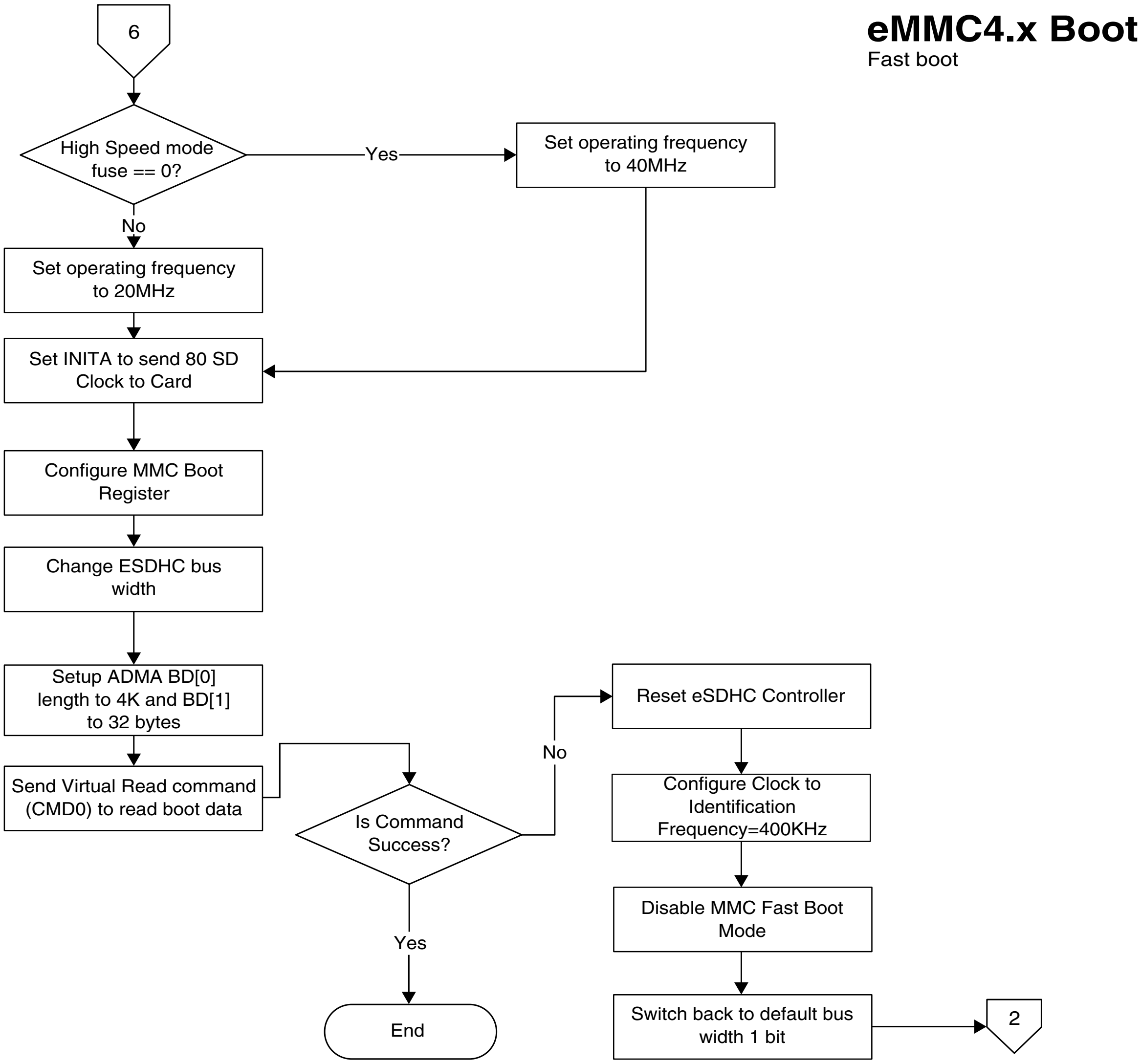


Figure 19-9. Expansion Device Boot Flow (2/7)

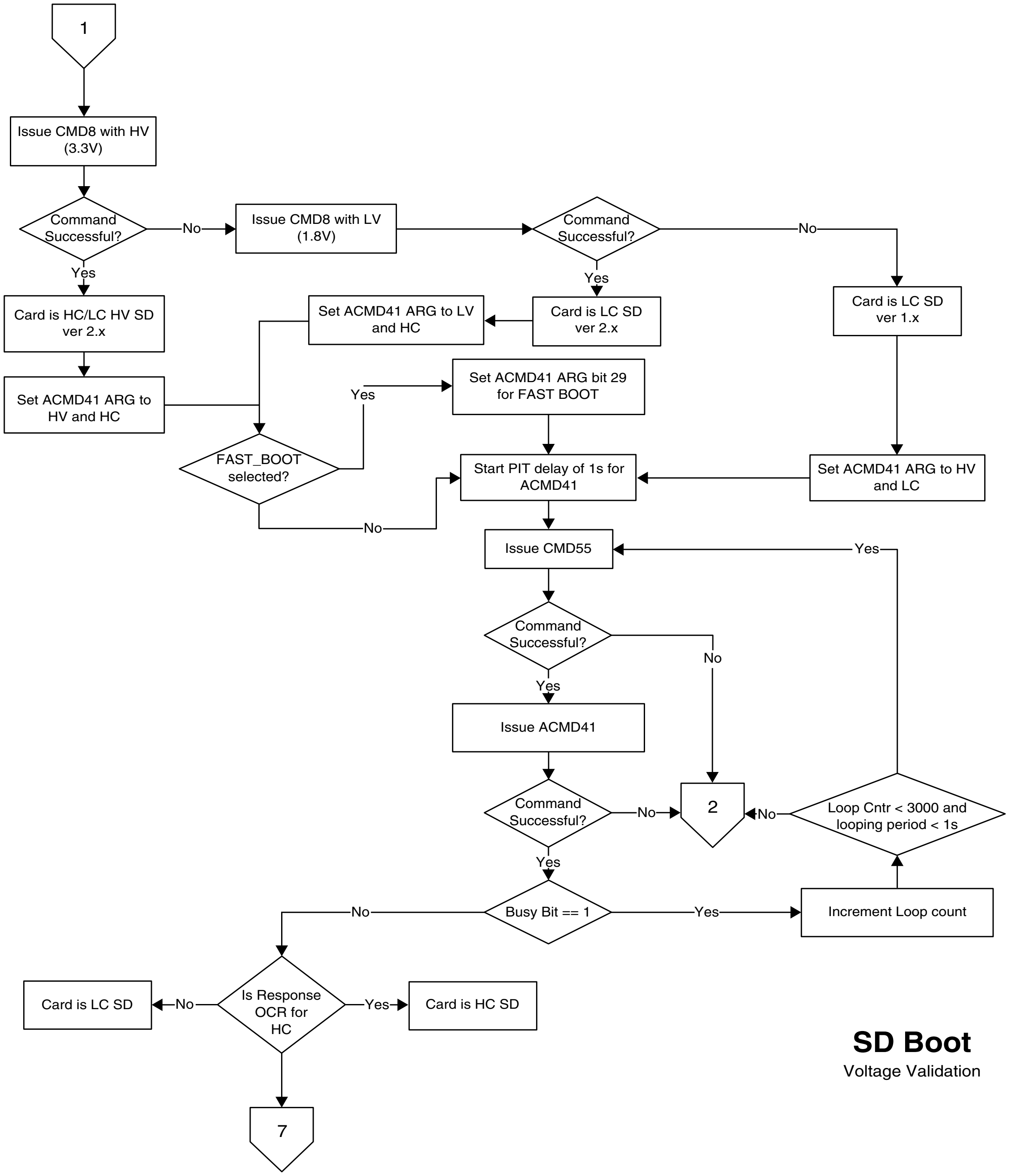


Figure 19-10. Expansion Device Boot Flow (3/7)

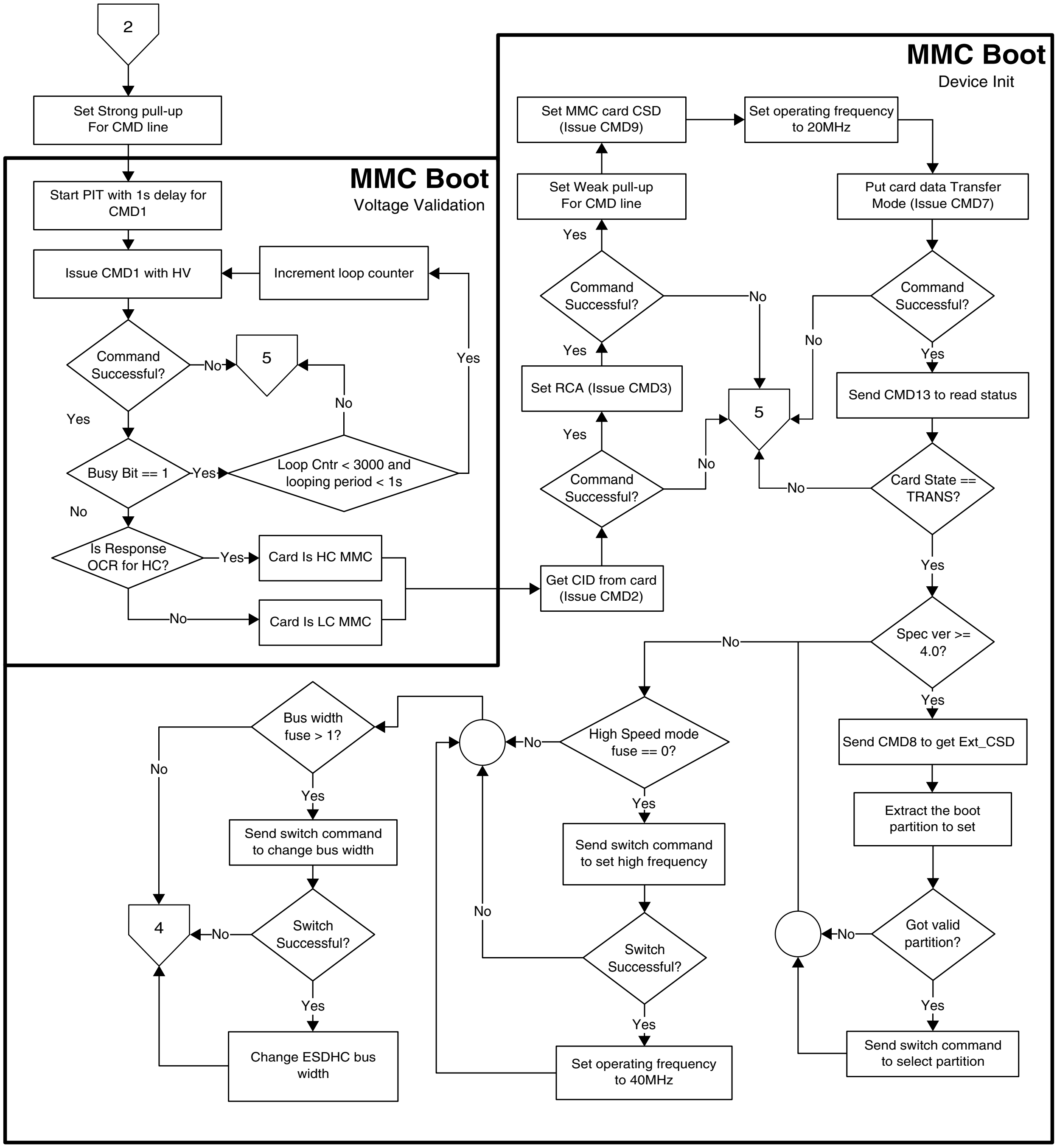


Figure 19-11. Expansion Device Boot Flow (4/7)

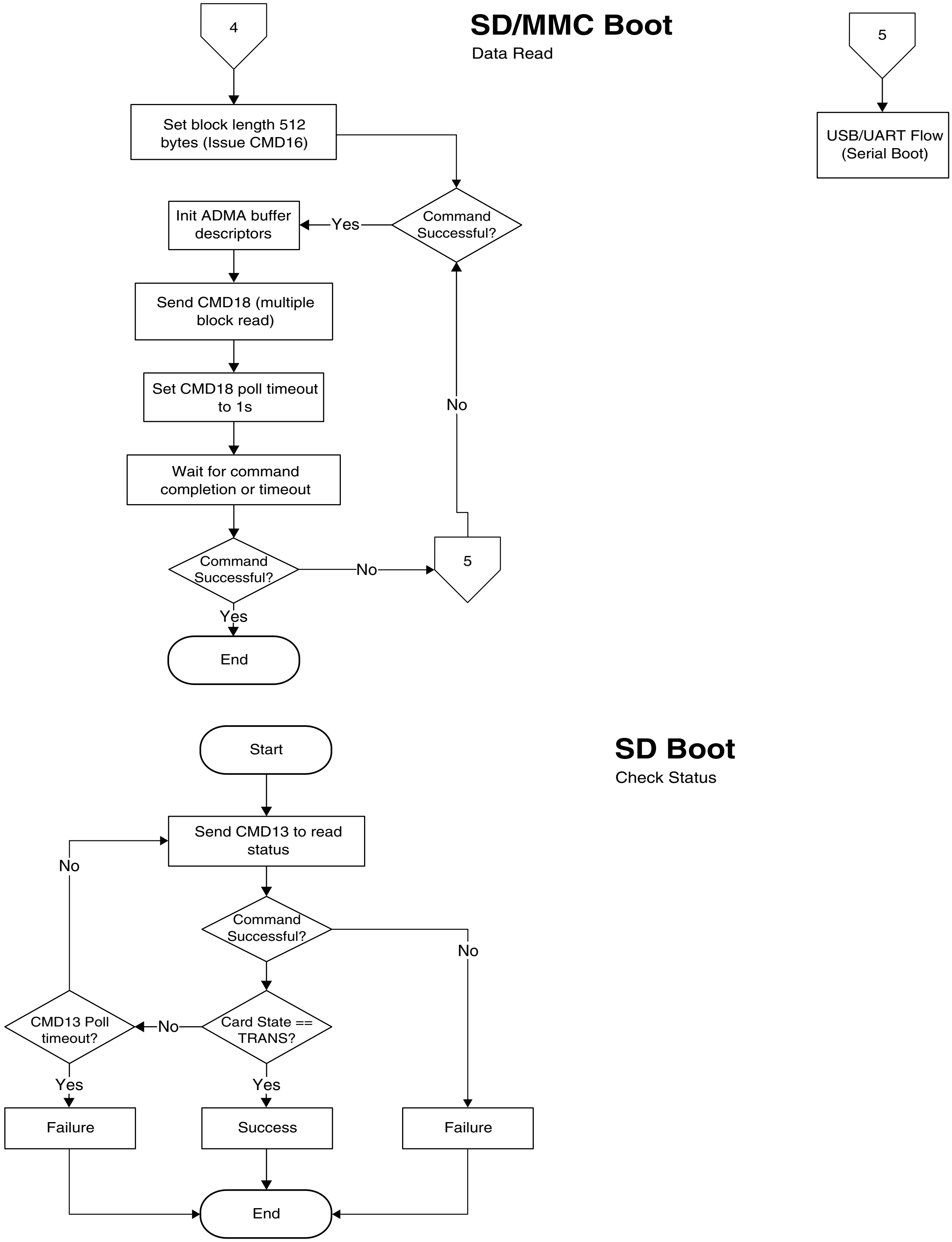


Figure 19-12. Expansion Device Boot Flow (5/7)

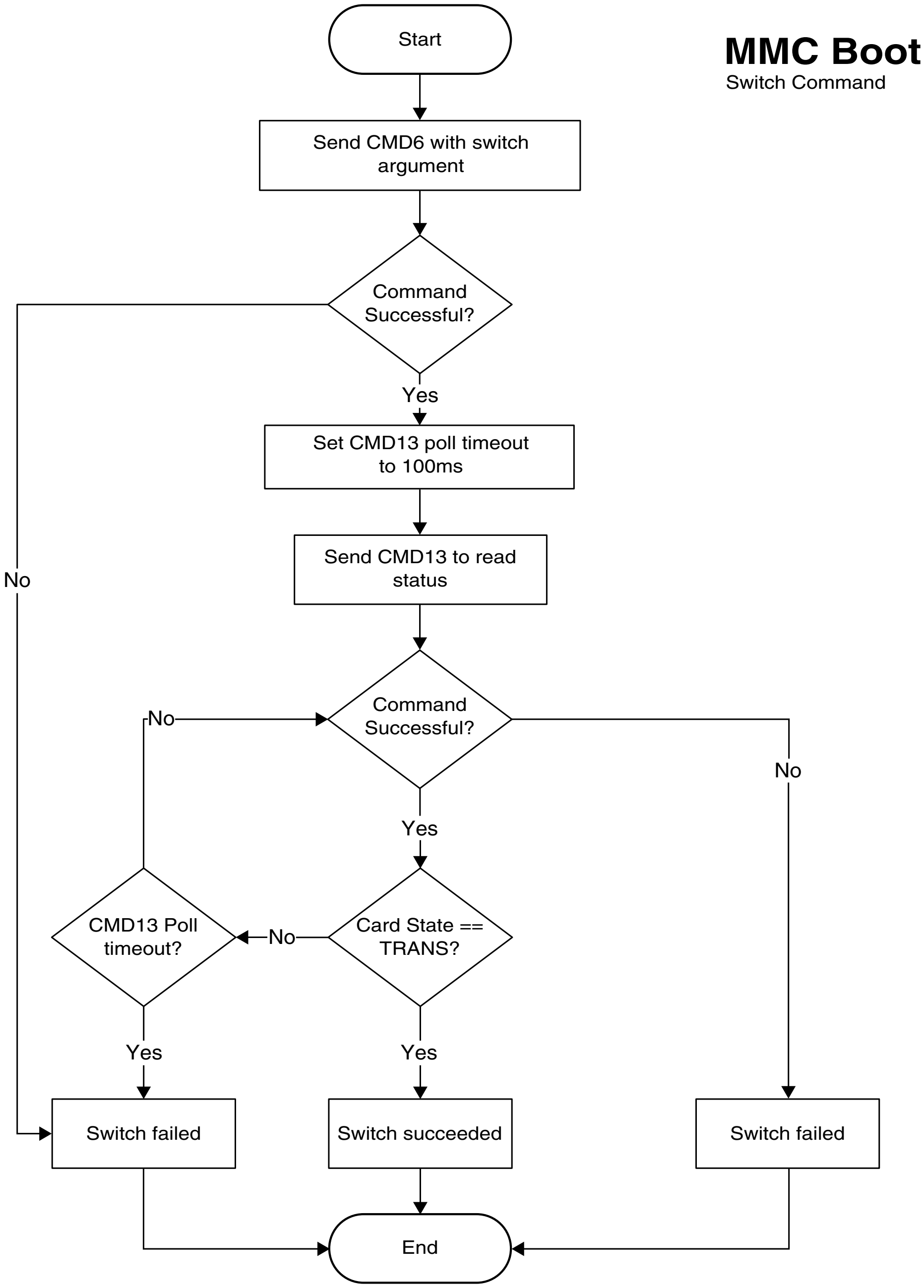


Figure 19-13. Expansion Device Boot Flow (6/7)

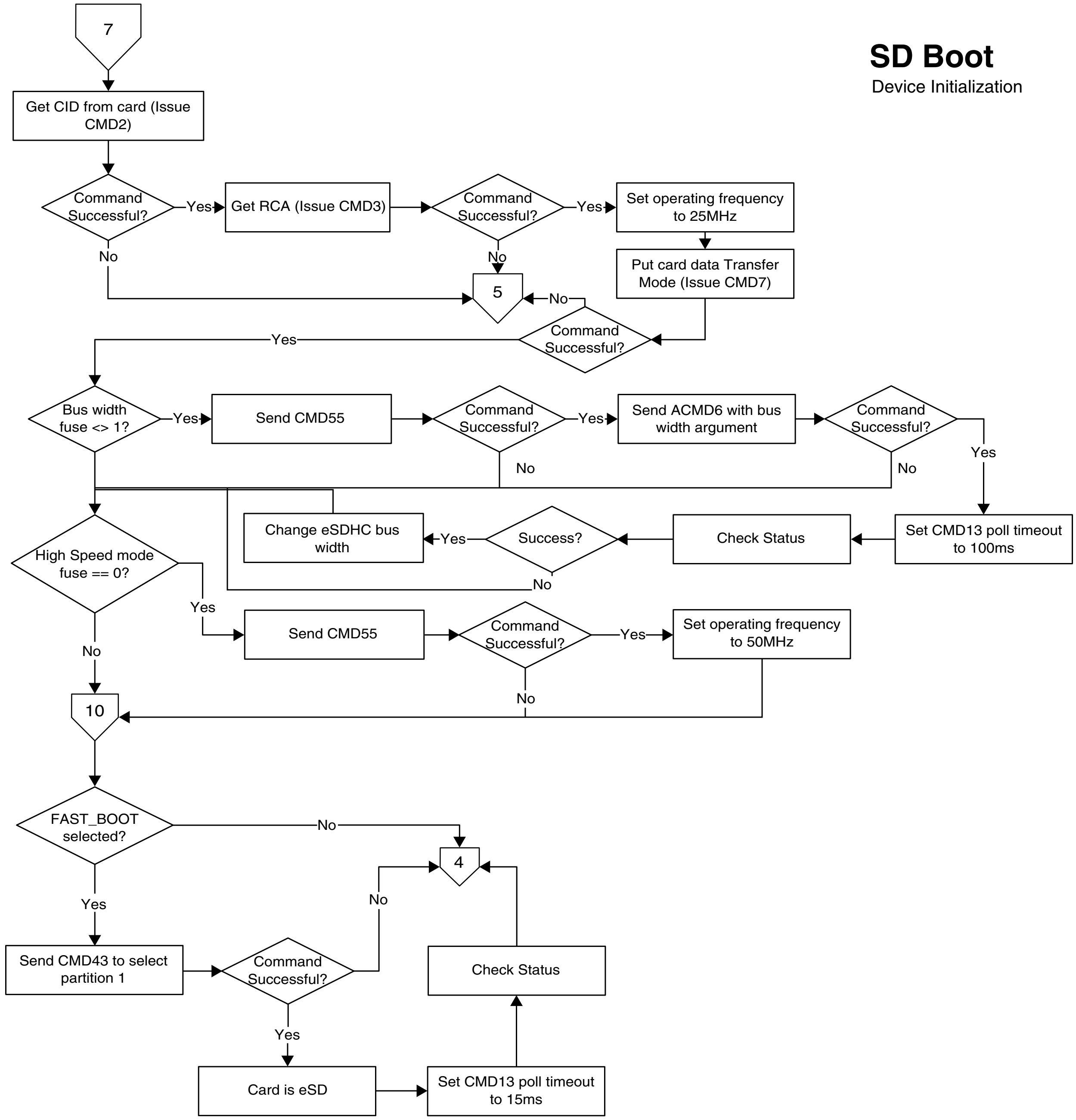


Figure 19-14. Expansion Device Boot Flow (7/7)

19.5.5.3 SD, eSD Boot

After the normal boot mode initialization begins, the SD/eSD frequency is set to 400 KHz. During the identification phase, SD/eSD card voltage validation is performed. During voltage validation, boot code first checks with high voltage settings; if that fails, it checks with low voltage settings. The capacity of the card is also checked. Boot code supports high capacity and low capacity SD/eSD cards after voltage validation card initialization is done.

During card initialization, the ROM boot code attempts to set the boot partition for all SD and eSD devices. If this fails, the boot code assumes the card is a normal SD card. If it does not fail, the boot code assumes it is an eSD card. After the initialization phase is over, boot code switches to a higher frequency (22 MHz in Normal Speed mode or 44 MHz in High Speed Mode).

The SD clock speed can be selected by BOOT_CFG1 [1].

19.5.5.4 IOMUX Configuration for SD/MMC

The table below shows ESDHC IOMUX pin configuration.

Table 19-27. ESDHC Pin Configuration

Signal	Pad Name	
	eSDHC0	eSDHC1
CLK	RMII0_MDC	TRACED[8]
CMD	RMII0_MDIO	TRACED[9]
DAT[0]	RMII0_CRS_DV	TRACED[10]
DAT[1]	RMII0_RXD[1]	TRACED[11]
DAT[2]	RMII0_RXD[0]	TRACED[12]
DAT[3]	RMII0_RXER	TRACED[13]
DAT[4]	FB_AD[23]	-
DAT[5]	FB_AD[22]	-
DAT[6]	FB_AD[21]	-
DAT[7]	FB_AD[20]	-

19.5.5.5 CCM Settings in various modes

Table 19-28. ESDHC0 CCM Settings

ESDHC0					
Registers	INIT	SD_LOW	SD_HIGH	MMC_LOW	MMC_HIGH
Clock Source	PLTF_BUS	PLL1_PFD3	PLL1_PFD3	PLL3_PFD3	PLL3_PFD3
CCSR	0x0004_0824	0x0004_0C24	0x0004_0C24	0x4004_0824	0x4004_0824
CSCMR1	0x0003_0000	0x0002_0000	0x0002_0000	0x0001_0000	0x0001_0000
CSCDR2	0x000A_0000	0x0008_0000	0x0008_0000	0x000F_0000	0x0006_0000

Table 19-29. ESDHC1 CCM Settings

ESDHC1					
Registers	**INIT**	**SD_LOW**	**SD_HIGH**	**MMC_LOW**	**MMC_HIGH**
Clock Source	PLTF_BUS	PLL1_PFD3	PLL1_PFD3	PLL3_PFD3	PLL3_PFD3
CCSR	0x0004_0824	0x0004_0C24	0x0004_0C24	0x4004_0824	0x4004_0824
CSCMR1	0x000C_0000	0x0008_0000	0x0008_0000	0x0004_0000	0x0004_0000
CSCDR2	0x00A0_0000	0x0080_0000	0x0080_0000	0x00F0_0000	0x0060_0000

19.5.5.6 Redundant Boot Support for Expansion Device

The device ROM supports redundant boot for expansion device. Primary or Secondary image is selected depending on PERSIST_SECONDARY_BOOT setting (see Table 19-7).

If PERSIST_SECONDARY_BOOT is 0, the boot ROM uses address 0x0 for primary image.

If PERSIST_SECONDARY_BOOT is 1, the boot ROM will read secondary image table from address 0x200 on boot media and will use address specified in the table.

Table 19-30. Secondary Image Table Format

Reserved (chipNum)
Reserved (driveType)
tag
firstSectorNumber
Reserved (sectorCount)

Where:

- tag: used as indication of valid secondary image table. Must be 0x00112233.
- firstSectorNumber is the first 512B sector number of the secondary image.

For secondary image support, the primary image must reserve space for secondary image table. See the figure below for typical structures layout on expansion device.

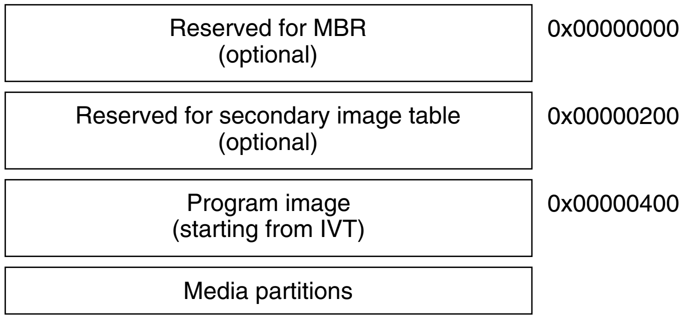


Figure 19-15. Expansion Device Structures Layout

For Closed mode, if there are failures during primary image authentication, the BOOTROM will set PERSIST_SECONDARY_BOOT bit (see Table 19-7) and perform software reset. (After software reset, secondary image will be used.)

For Unsecured chip, if invalid Image Vector Table is read, then BOOTROM will set PERSIST_SECONDARY_BOOT bit (see Table 19-7) and perform software reset. (After software reset, secondary image will be used.)

NOTE

PERSIST_SECONDARY_BOOT bit can be cleared by application after boot or will be cleared automatically during Power On Reset.

19.5.6 NAND Flash Boot using NFC Interface

A number of MLC/SLC NAND Flash devices from different vendors are supported by the boot ROM. The Error Correction and Control (ECC) sub-block is used to detect the errors.

19.5.6.1 NAND Flash eFUSE Configuration

The boot ROM determines the configuration of external the NAND flash by parameters, either provided by eFUSE, or sampled on GPIO pins, during boot. See the table below for parameters details.

Table 19-31. NAND Boot eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[7]	OEM	Boot Device Selection	Yes	0	1 – BOOT from Nand Interface

Table continues on the next page...

Table 19-31. NAND Boot eFUSE Descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG1[6]	OEM	Nand Interface Frequency Select	Yes	0	1 – Normal Mode 33MHz 0 – Fast Mode 40MHz
BOOT_CFG1[5]	OEM	Fast Boot Acknowledge	Yes	0	1 – Fast flash timing 0 – Slow flash timing
BOOT_CFG1[3]	OEM	Nand Chip Enable	Yes	0	0 – Use Chip Enable 0 1 – Use Chip Enable 1
BOOT_CFG1[2]	OEM	Nand Interface Data Width	Yes	0	0 –8 Bit Interface 1 – 16 Bit interface
BOOT_CFG1[1:0]	OEM	Address Cycles	Yes	00	00 – 3 Address Cycles 01 – 2 Address Cycles 1X – Reserved
BOOT_CFG2[4:3]	OEM	Boot Search Count	Yes	00	0X – 2 10 – 4 11 – 8
BOOT_CFG2[2:1]	OEM	Pages in a Block	Yes	00	00 – 128 01 – 64 10 – 32 11 – Reserved
BOOT_CFG2[0]	OEM	Override Pad Settings	Yes	0	0 – use Default Settings 1 – Use Pad setting values
NAND_READ_CMD_CODE1[7:0][2]	OEM	Read Command Code 1	No	0	Read Command Code 1
NAND_READ_CMD_CODE2[7:0][3]	OEM	Read Command Code 2	No	0	Read Command Code 2

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 19-3 for corresponding GPIO pin.
2. Bank0 Word7 Field[7:0]
3. Bank0 Word7 Field[15:8]

NOTE

Nand Flash Read Command Codes are read from fuses. If Command Code 2 is 0x00 then Command Code 0x30 is used in BOOTROM.

19.5.6.2 NAND Flash Boot Flow and BOOT Control Blocks (BCB)

There are two BCB data structures: FCB and DBBT. As part of the NAND media initialization, the ROM driver searches for a Firmware Configuration Block (FCB) that contains the page address of Discovered Bad Block Table (DBBT) Search Area and start page address of primary and secondary firmware.

The hardware ECC level to use is embedded inside FCB block. The FCB data structure is itself protected using software ECC (SEC-DED Hamming Codes). Driver reads raw 2112 bytes of first sector and runs through software ECC engine that determines whether FCB data is valid or not.

If FCB is not found or ECC fails, or the fingerprints do not match, the Block Search state machine increments page number to Search Stride number of pages to read for the next BCB until SearchCount pages have been read.

If search fails to find a valid FCB, the NAND driver responds with an error and the BOOTROM enters into serial download mode.

The FCB contains the page address of DBBT Search Area, and the page address for primary and secondary boot images. DBBT is searched in DBBT Search Area just like how FCB is searched. After the FCB is read, the DBBT is loaded, and the primary or secondary boot image is loaded using starting page address from FCB.

See the flow of FCB search in the figure below.

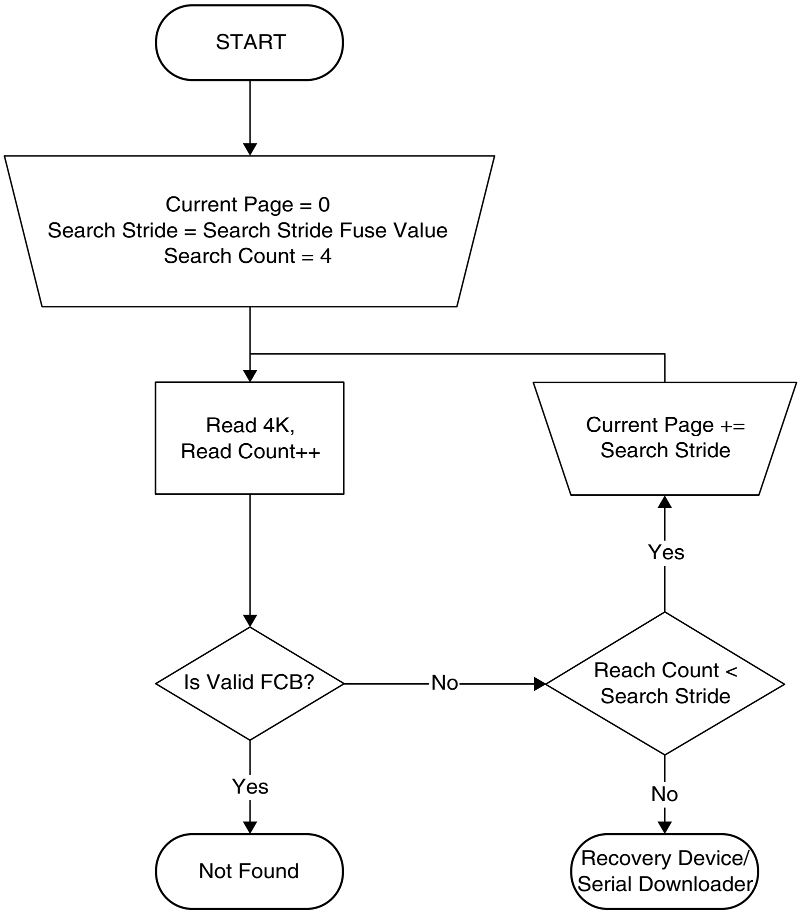


Figure 19-16. FCB Search Flow

Once FCB found the BOOTROM starts searching for Discovered Bad Blocks Table (DBBT). If DBBT Search Area is 0 in FCB, then ROM assumes that there are no bad blocks on NAND device. See the figure below for DBBT search flow.

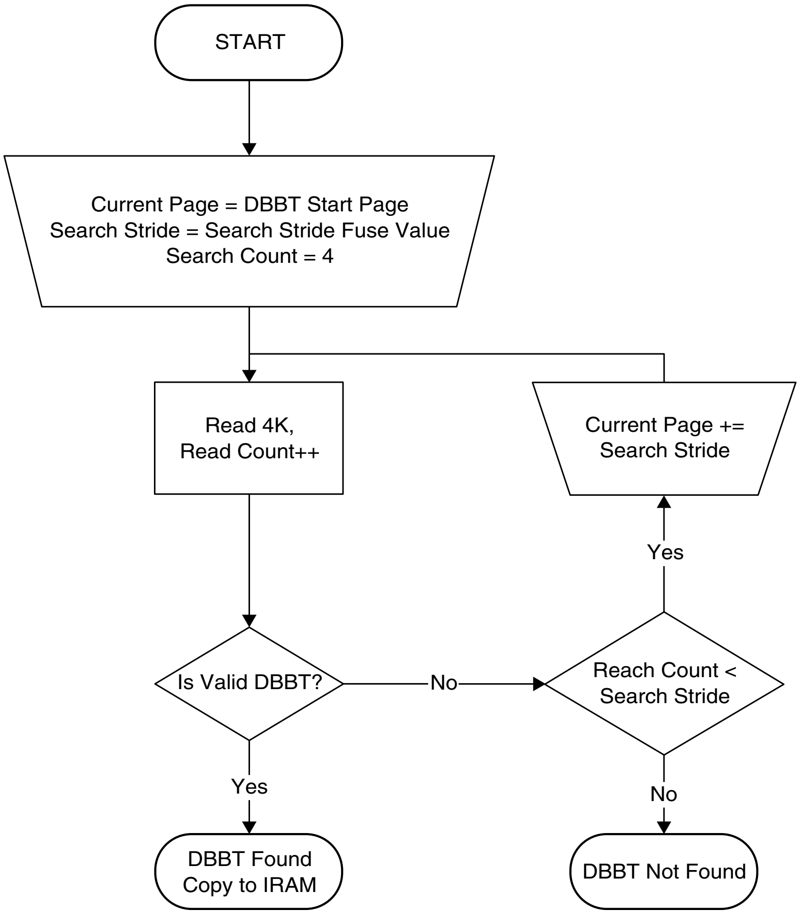


Figure 19-17. DBBT Search Flow

If during primary image read there was a page with number of errors higher than ECC can correct, the BOOTROM will turn on PERSIST_SECONDARY_BOOT bit and perform SW reset. (After SW reset secondary image will be used).

If during secondary image read there was a page with number of errors higher than ECC can correct, the BOOTROM will go to serial loader.

NOTE

PERSIST_SECONDARY_BOOT bit can be cleared by application after boot or will be cleared automatically during Power On Reset.

19.5.6.3 Firmware Configuration Block

The FCB is the first sector in the first good block. The FCB should be present at each search stride of the search area. The search area contains copies of the FCB at each stride distance, so in case the first NAND block becomes corrupted, the ROM will find its copy

in the next NAND block. The search area should span over at least two NAND blocks. The location information for DBBT search area, FW1, and FW2 are all specified in the FCB. Flash Control Block Structure is as shown in the table below.

Table 19-32. Flash Control Block Structure

Name	Start Byte	Size in Bytes	Description
Reserved	0	4	Reserved for Fingerprint #1(Checksum)
FingerPrint	4	4	32 bit word with a value of 0x46434220, in ASCII "FCB"
Version	8	4	32-bit version number; this version of FCB is 0x00000001
Reserved	12	8	Not used by ROM
DataPageSize	20	4	Number of bytes of data in a page. Typically, this is 2048 bytes for 2112 bytes page size or 4096 bytes for 4314 bytes page size
TotalPageSize	24	4	Total number of bytes in page. Typically, 2112 for 2 K page or 4224 or 4314 for 4 K page.
SectorsPerBlock	28	4	Number of pages per block. Typically 64 or 128 or depending on NAND device type.
Reserved	32	24	Not used by ROM
EccBlock0EccType	56	4	Value from 0 to 7 used to set Error Correction Type for Block of ECC page
Reserved	60	44	Not used by ROM
Firmware1_startingSector	104	4	Page number address where first copy of bootable firmware is located
Firmware2_startingSector	108	4	Page number address where second copy of bootable firmware is located
Reserved	112	8	Not used by ROM
DBBTSearchAreaStartAddress	120	4	Page address for bad block table search area
Reserved	124	8	Not used by ROM
BBMarkerPhysicalOffset	132	4	This is the offset where manufacturer leaves bad block marker on a page
Reserved	136	36	Not used by ROM
DISBBM	172	4	If 0 ROM will swap BadBlockMarkerByte with SpareArea[0] after reading a page. If the value set is 1 then ROM will not do swapping
Reserved	176	40	Not used by ROM
DISBB_Search	216	4	Disable the bad block search function when reading the firmware, only using DBBT.
Bad Block Search Limit	220	4	Number of Blocks to be searched when a Bad Block is found. By Default the value of this field is 2.

NOTE

Firmware1_startingSector, Firmware2_startingSector and DBBTSearchAreaStartAddress must point to a page in a good

Block. For performing Bad Block marker check, firmware 1 and 2 starting sectors must be block aligned.

19.5.6.4 Discovered Bad Block Table

See the table below for DBBT format.

Table 19-33. DBBT Structure

Name	Start Byte	Size in Bytes	Description
Reserved	0	4	-
FingerPrint	4	4	32-bit word with a value of 0x44424254,in ASCII "DBBT"
Version	8	4	32-bit version number; this version of DBBT is 0x00000001
Reserved	12	4	-
DBBT_NUM_OF_PAGES	16	4	Size of DBBT in pages
Reserved	20	4*PageSize -20	-
Reserved	4*PageSize	4	-
Number of Entries	4*PageSize + 4	4	Number of Bad Blocks
Bad Block Number	4*PageSize + 8	4	First Bad Block number
Bad Block Number	4*PageSize + 12	4	Second Block number
…..	….	….	….
…..	….	….	….
Last Bad Block Number	….	….	Last Bad Block number

19.5.6.5 Typical NAND Page Organization

Nand Page is divided into multiple data blocks each of size 2048 bytes. Figure below shows a typical layout of NAND page of size 4224 bytes.

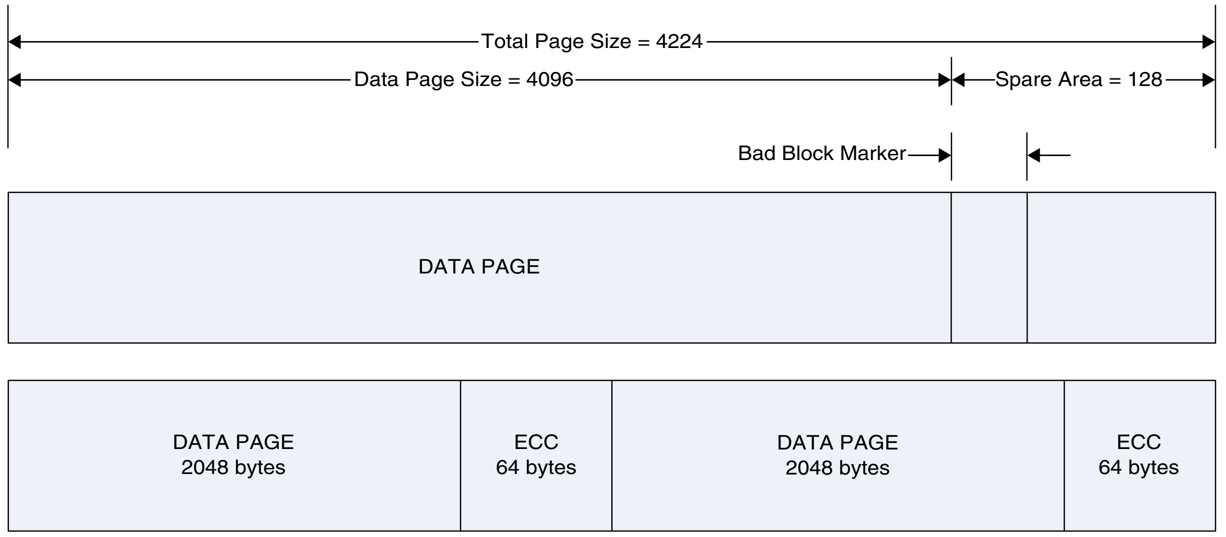


Figure 19-18. Valid Layout for 4224 bytes Sized Page

There are 2 data blocks of 2048 bytes each in a page of a 4k page sized NAND. The values of ECC should be such that above calculation should not result in a value greater than the size of a page in a 4k page NAND.

For NAND Boot, with page size restriction and data block size restricted to 2048 bytes, only a few combinations of ECC for block.

19.5.6.6 Bad Block Handling in the ROM

During firmware boot, at the block boundary, the Bad Block table is searched for a match to the next block. If no match is found, the next block can be loaded. If a match is found, the block must be skipped and the next block checked.

If Bad Block table start page is null, check the manufactory made Bad Block marker. The location of Bad Block maker is at the first 3 or last 3 pages in every block of the NAND flash. Nand manufacturers normally use one byte in the spare area of certain pages within a block to mark a block is bad or not. 0xFF means good block, otherwise means bad block.

In order to preserve the BI (bad block information), flash updater or gang programmer applications need to swap Bad Block Information (BI) data (1 byte in length) to appropriate place in DATA Area for every page before programming NAND flash.

This is calculated using the following formula.

Total Page Size -> T_{NAND} bytes

Data Page Size -> D_{NAND} bytes

Bad Block Marker Offset from start of Page -> BB_{Offset}

Total Spare Area Size -> $TSP_{NAND} = T_{NAND} - D_{NAND}$

No of Virtual Pages -> $NVP = D_{NAND} / 2048$

Spare Area Page Size per Virtual Page -> $VSP_{NAND} = TSP_{NAND} /NVP$

Virtual Page Size -> $VP_{Size} = 2048 + VSP_{NAND}$

Bad Block Marker Offset in Virtual Page ($BBOFF_{VP}$)

$BBOFF_{VP} = MOD(BB_{Offset}, VP_{Size})$

Bad Block Marker in Virtual Page count ($BBOFF_{NVP}$)

$BBOFF_{NVP} = BB_{Offset} / VP_{Size}$

ROM when loading firmware copies back 1 byte from the Virtual Page to BI offset in page data. The figure below shows how the factory bad block marker is preserved.

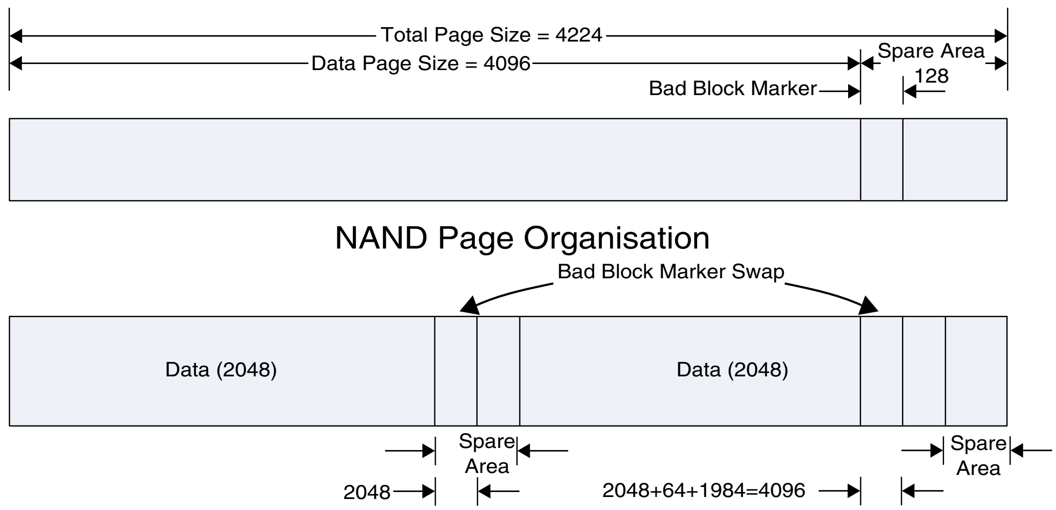


Figure 19-19. Factory Bad Block Marker Preservation

In the FCB structure, there is BBMarkerPhysicalOffset to indicate Bad Block Marker place in the page data, that manufacturer uses to mark bad block.

Table Below show typical values for different NAND Page Sizes:

Table 19-34. Bad Block Marker Position in NAND Page Layout

T_{NAND}	D_{NAND}	BB_{OFFSET}	TSP_{NAND}	NVP	VSP_{NAND}	VP_{SIZE}	$BBOFF_{VP}$	$BBOFF_{NVP}$
2112	2048	2048	64	1	64	2112	2048	0
4224	4096	4096	128	2	64	2112	1984	1
4304	4096	4096	208	2	104	2152	1944	1
8640	8192	8192	448	4	112	2160	1712	3
NOTE: Data Page Size = 2048 bytes								

19.5.6.7 Setup DMA for DDR Transfers

In DMA descriptors NAND Flash Controller is configured to read page data at double data rate, the word length is set to 16 and transfer count to half of page size.

19.5.6.8 IOMUX Configuration for NFC

The table below shows the Nand Flash Controller IOMUX pin configuration.

Table 19-35. NAND IOMUX Pin Configuration

Signal	Pad Name
NF_ALE	SAI1_RX_BCLK
NF_CLE	SAI1_RX_DATA
NF_CE0_b	SAI0_RX_DATA
NF_CE1_b	SAI0_TX_DATA
NF_IO[15]	FB_AD[31]
NF_IO[14]	FB_AD[30]
NF_IO[13]	FB_AD[29]
NF_IO[12]	FB_AD[28]
NF_IO[11]	FB_AD[27]
NF_IO[10]	FB_AD[26]
NF_IO[9]	FB_AD[25]
NF_IO[8]	FB_AD[24]
NF_IO[7]	FB_AD[23]
NF_IO[6]	FB_AD[22]
NF_IO[5]	FB_AD[21]
NF_IO[4]	FB_AD[20]
NF_IO[3]	FB_AD[19]
NF_IO[2]	FB_AD[18]
NF_IO[1]	FB_AD[17]
NF_IO[0]	FB_AD[16]
NF_R/B_b	SAI1_TX_BCLK
NF_RE_b	SAI0_RX_SYNC
NF_WE_b	SAI0_RX_BCLK

19.5.6.9 CCM Settings in various modes

Table 19-36. NFC CCM Settings

NFC		
Resgisters	**Normal Mode**	**Fast Mode**
Clock Source	PLL3_PFD1	PLL3_PFD1
CCSR	0x1004_0824	0x1004_0824
CSCMR1	0x0000_0200	0x0000_0200
CSCDR2	0x0000_2010	0x0000_2010
CSCDR3	0x0000_6000	0x0000_4000

19.6 Program Image

This section describes the data structures that are required to be included in a user's Program Image. A Program Image consists of:

- Image Vector Table- A list of pointers located at a fixed address that the ROM examines to determine where other components of the Program Image are located.
- Boot Data- A table indicating the Program Image Location, Program Image size in bytes and plugin flag.
- Device Configuration Data- IC configuration data.
- Data used by HAB- CSF command data, signatures and certificates
- User code and data

19.6.1 Image Vector Table and Boot Data

The Image Vector Table (IVT) is the data structure that the ROM reads from the boot device supplying the program image containing the required data components to perform a successful boot. The IVT includes the program image entry point, a pointer to Device Configuration Data (DCD) and other pointers used by the ROM during the boot process. The ROM locates the IVT at a fixed address that is determined by the boot device connected to Vybrid. The IVT offset from the base address and initial load region size for each boot device type is defined in the table below. The location of the IVT is the only fixed requirement by the ROM. The remainder or the image memory map is flexible and is determined by the contents of the IVT.

Table 19-37. Image Vector Table Offset and Initial Load Region Size

Boot Device Type	Image Vector Table Offset	Initial Load Region Size
NOR	4 Kbyte = 0x1000 bytes	Entire Image Size
NAND	1 Kbyte = 0x400 bytes	4 Kbyte
SD/MMC/eSD/eMMC	1 Kbyte = 0x400 bytes	4 Kbyte
I2C/SPI EEPROM	1 Kbyte = 0x400 bytes	4 Kbyte
Quad SPI	1 Kbyte = 0x400 bytes	Entire Image Size

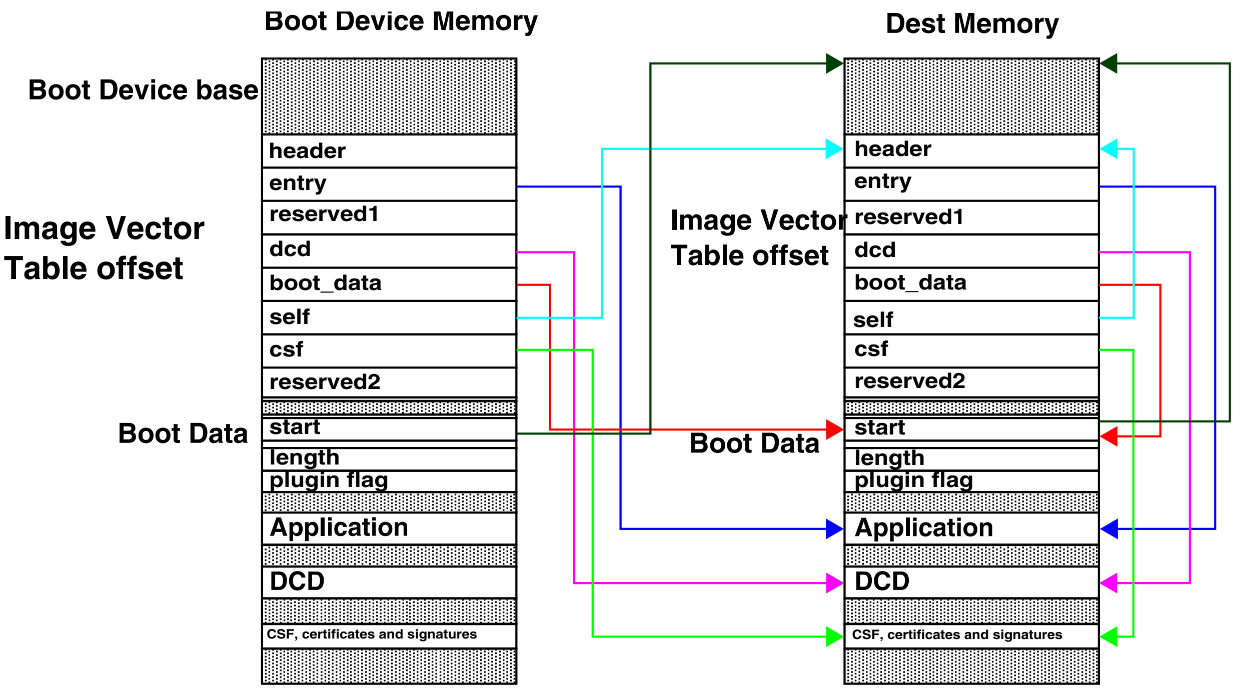


Figure 19-20. Image Vector Table

19.6.1.1 Image Vector Table Structure

The IVT has the following format where each entry is a 32 bit word:

Table 19-38. IVT Format

Header (See table 30 below)
entry: Absolute address of the first instruction to execute from the image
reserved1: Reserved and should be zero
dcd: Absolute address of the image DCD. The DCD is optional so this field may be set to NULL if no DCD is required. See Device Configuration Data (DCD) for further details on DCD.
boot data: Absolute address of the Boot Data
self: Absolute address of the IVT. Used internally by the ROM

Table continues on the next page...

Table 19-38. IVT Format (continued)

csf: Absolute address of Command Sequence File (CSF) used by the HAB library. See High Assurance Boot (HAB) for details on secure boot using HAB. This field must be set to NULL when not performing a secure boot
reserved2: Reserved and should be zero

The IVT header has the following format:

Table 19-39. IVT Header Format

Tag	Length	Version

Where:

```
Tag: A single byte field set to 0xD1
Length: a two byte field in big endian format containing the overall length of the IVT, in
bytes, including the header. (The length is fixed and must have a value of 32 bytes)
Version: A single byte field set to 0x41
```

19.6.1.2 Boot Data Structure

The Boot Data must follow the format defined in the table below; each entry is a 32-bit word.

Table 19-40. Device BOOT Data Format

start	Absolute Address of Image
length	Size of Program Image
plugin	Plug-in Flag

19.6.2 Device Configuration Data (DCD)

Upon reset, the device uses the default register values for all peripherals in the system. However, these settings typically are not ideal for achieving optimal system performance and there are even some peripherals that must be configured before they can be used. The DCD is configuration information contained in a Program Image, external to the ROM that the ROM interprets to configure various peripherals on this device. Some components such as SDRAM require some sequence of register programming as part of configuration before it is ready to be used. The DCD feature can be used to program DDRMC registers to the optimal settings.

The ROM determines the location of the DCD table based on information located in the Image Vector Table (IVT). See Image Vector Table and Boot Data for more details. The DCD table shown below is a big endian byte array of the allowable DCD commands. The maximum size of the DCD limited to 3028 bytes.

Table 19-41. DCD Data Format

Header
[CMD]
[CMD]
[CMD]
….

The DCD header is 4 bytes with the following format:

Table 19-42. DCD Header

Tag	Length	Version

Where:

```
Tag: A single byte field set to 0xD2
Length: a two byte field in big endian format containing the overall length of the DCD, in
bytes, including the header.
Version: A single byte field set to 0x40
```

19.6.2.1 Write Data Command

The Write Data Command is used to write a list of given 1, 2 or 4-byte values or bitmasks to a corresponding list of target addresses. The format of Write Data Command, again a big endian byte array, is shown in the table below.

Table 19-43. Write Data Command Format

Tag	Length	Parameter
Address		
Value/Mask		
[Address]		
[Value/Mask]		
….		
[Address]		
[Value/Mask]		

Where:

```
Tag: A single byte field set to 0xCC
Length: A two byte field in big endian format containing the length of the Write Data
Command, in bytes, including the header
Address: target address to which data should be written
Value/Mask: data value or bitmask to be written to preceding address
```

The Parameter field is a single byte divided into bitfields as follows:

Table 19-44. Write Data Command Parameter field

7	6	5	4	3	2	1	0
flags					bytes		

Where:

```
bytes: width of target locations in bytes. Either 1, 2 or 4
flags: control flags for command behavior.
Data Mask = bit 3: if set, only specific bits may be overwritten at target address
(otherwise all bits may be overwritten
Data Set = bit 4: if set, bits at the target address overwritten with this flag
(otherwise it is ignored)
```

One or more target address and value/bitmask pairs can be specified. The same bytes and flags parameters apply to all locations in the command.

When successful, this command writes to each target address in accordance with the flags as follows:

Table 19-45. Interpretation of Write Data Command Flags

"Mask"	"Set"	Action	Interpretation
0	0	*address = val_msk	Write Value
0	1	*address = val_msk	Write Value
1	0	*address &= ~val_msk	Clear bitmask
1	1	*address |= val_msk	Set bitmask

Note

If any of the target addresses does not have the same alignment as the data width indicated in the parameter field, none of the values are written.

If any of the values is larger or any of the bitmask is wider than permitted by the data width indicated in the parameter field, none of the values are written.

If any of the target addresses do not lie within an allowed region, none of the values are written. The list of blocks and target addresses allowed for this device are given below.

Table 19-46. Valid DCD Memory Address Ranges

Memory Region	Address Range	Cortex-A5	Cortex-M4
SDRAM0	0x8000_0000 - 0xDFFF_FFFF	YES	YES
TCML	0x1F80_0000 - 0x1FFF_FFFF	YES	YES
QSPI0	0x2000_0000 - 0x2FFF_FFFF	YES	YES
FlexBUS	0x3000_0000 - 0x3EFF_FFFF	YES	YES
OCRAM - Sys RAM	0x3F00_0000 - 0x3F3F_FFFF	YES	YES
OCRAM - gfx RAM	0x3F40_0000 - 0x3F7F_FFFF	YES	YES
TCMU	0x3F80_0000 - 0x3FFF_FFFF	YES	YES
QSPI1	0x5000_0000 - 0x5FFF_FFFF	YES	YES
SDRAM1	0xE000_0000 - 0xFFFF_FFFF	YES	NO
Cortex-M4 Code Alias Region	0x0080_0000 - 0x1F7F_FFFF	NO	YES

Table 19-47. Valid DCD Peripheral Address Ranges

Peripheral	Address Range	Cortex-A5	Cortex-M4
DDRMC	0x400A_E000 - 0x400A_EFFF	YES	YES
IOMUX	0x4004_8000 - 0x4004_8FFF	YES	YES
FlexBUS Controller	0x4001_E000 - 0x4001_EFFF	YES	YES
Clock Control Module	0x4006_B000 - 0x4006_BFFF	YES	YES
ANADIG	0x4005_0000 - 0x4005_0FFF	YES	YES

19.6.2.2 Check Data Command

The Check Data Command is used to test for a given 1, 2 or 4-byte bitmasks from a source address. The Check Data Command is a big endian byte array with format shown in the table below.

Table 19-48. Check Data Command Format

<table>
<tr><td>Tag</td><td>Length</td><td>Parameter</td></tr>
<tr><td colspan="3">Address</td></tr>
<tr><td colspan="3">Mask</td></tr>
<tr><td colspan="3">[Count]</td></tr>
</table>

Where:

```
Tag: A single byte field se to 0xCF
Length: A two byte field in big endian format containing the length of the Check Data
Command, in bytes, including the header
Address: source address to test
Mask: bit mask to test
Count: optional poll count. If count is not specified this command will poll indefinitely
until the exit condition is met. If count = 0, this command behaves as for NOP.
```

The Parameter field is a single byte divided into bitfields as follows:

Table 19-49. Check Data Command Parameter field

7	6	5	4	3	2	1	0
flags				bytes			

Where:

```
bytes: width of target locations in bytes. Either 1, 2 or 4
flags: control flags for command behavior.
Data Mask = bit 3: if set, only specific bits may be overwritten at target address
(otherwise all bits may be overwritten)
Data Set = bit 4: if set, bits at the target address overwritten with this flag
(otherwise it is ignored)
```

This command polls the source address until either the exit condition is satisfied, or the poll count is reached. The exit condition is determined by the flags as follows:

Table 19-50. Interpretation of Check Data Command Flags

"Mask"	"Set"	Action	Interpretation
0	0	(*address & mask) == 0	All bits clear
0	1	(*address & mask) == mask	All bits set
1	0	(*address & mask) != mask	Any bit clear
1	1	(*address & mask) != 0	Any bit set

Note

If the source address does not have the same alignment as the data width indicated in the parameter field, the value is not read.

If the bitmask is wider than permitted by the data width indicated in the parameter field, the value is not read.

19.6.2.3 NOP Command

This command has no effect. The format of NOP Command is a little endian four byte array as shown in table below.

Table 19-51. NOP Command Format

Tag	Length	Undefined

Where:

```
Tag: A single byte field se to 0xC0
Length: A two byte field containing the length of the NOP Command in bytes. Fixed to a value
of 4.
Undefined: This byte is ignored and can be set to any value.
```

19.7 Plugin Image

The device ROM supports a limited number of boot devices. To boot from other devices (for example, Ethernet), a supported boot device must be used (typically serial ROM) that includes firmware for the desired boot device.

The Boot ROM will detect the image type by using the plugin flag in the Boot Data Structure. If the plugin flag is 1, then the ROM will use the image as a plugin function. The function must initialize the boot device and copy the program image to the final location. At the end, the plugin function must return with the parameters of the program image.

The boot ROM will authenticate the plugin image prior to running the plugin function and then will authenticate the program image.

The plugin function must follow the API described below:

```
typedef unsigned char (*) plugin_download_f(void **start, size_t *bytes, UINT32 *ivt_offset)
```

ARGUMENTS PASSED:

- start - Image load address on exit.

- bytes - Image size on exit.
- ivt_offset - Offset in bytes of the IVT from the image start address on exit.

RETURN VALUE:

- 1 - on success
- 0 - on failure

19.8 Serial Downloader (BOOT_MODE [1:0] = 0b01)

The Serial Downloader provides a means to download a Program Image to the device over the USB/UART connection. In this mode, the ROM programs either WDOG-A5 or WDOG-M4 for a 32-second time-out if WDOG_ENABLE eFUSE is 1, and then continuously polls for USB connection and UART activity. If no activity is found on USB or UART and the watchdog timer expires, then the ARM core is reset.

Note

The downloaded application image must continue to service the watchdog timer to avoid an undesired reset from occurring.

The USB/UART boot flow is shown in the figure below.

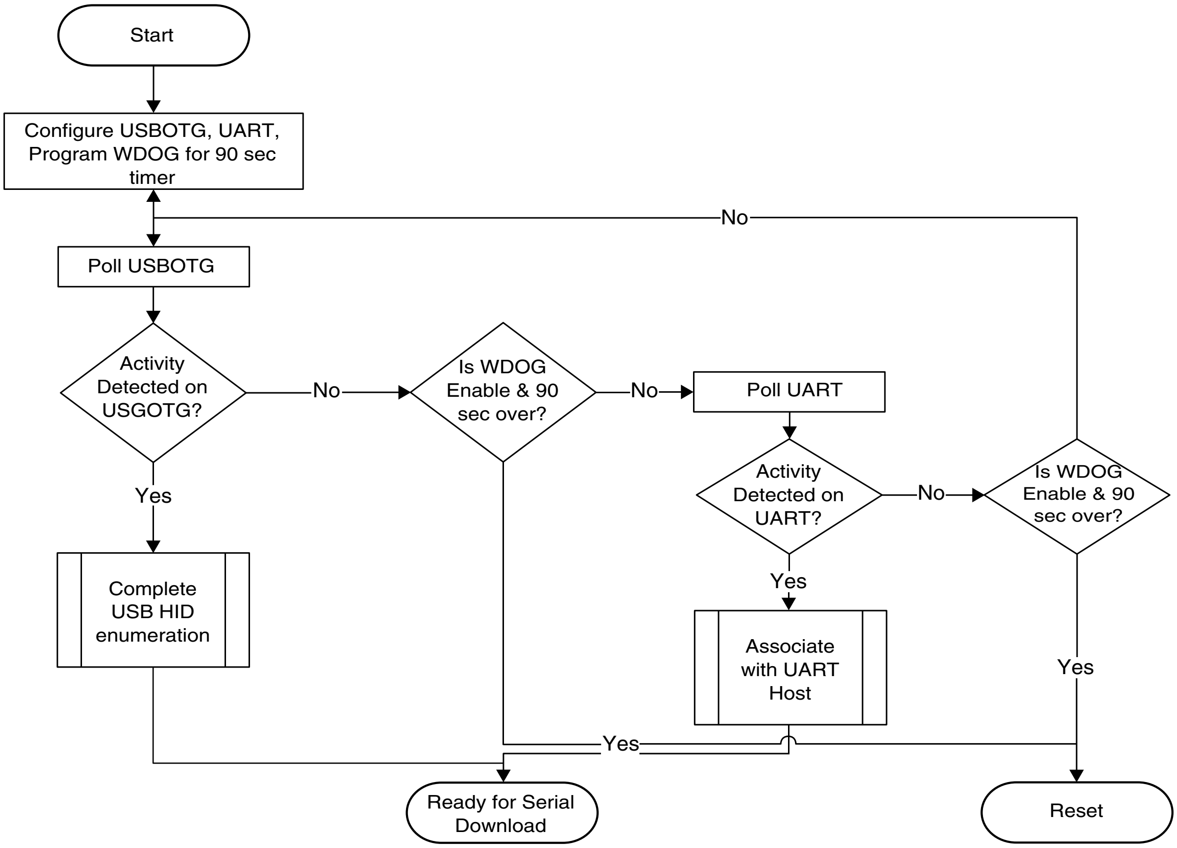


Figure 19-21. USB/UART Boot Flow

19.8.1 USB Boot Flow

USB support in this device is composed of the USB OTG core controller, compliant with the USB 2.0 specification and the USBPHY (HS USB transceiver). The device ROM supports the USB0 port for boot purposes. The other USB port on the device is not supported for boot purposes.

NOTE

Only USB0 is supported as a boot device. USB1 cannot be used.

The device USB Driver is implemented as a USB HID class. A collection of 4 HID reports are used to implement SDP protocol for data transfers:

- Report 1 transfer over Control OUT endpoint, 16-byte SDP command from host to device. Total bytes are 17 with first byte reserved for Report ID set to 1.

- Report 2 transfer over control OUT endpoint from host to device data associated with report 1 SDP command. Max report size is 1025 bytes with first byte reserved for Report ID set to 2.
- Report 3 transfer over interrupt endpoint from device to host to send HAB security state. Max size of this report is 5 bytes with first byte reserved for Report ID set to 3.
- Device sends 0x12343412 in production mode and 0x56787856 in engineering mode in the remaining 4 bytes of this report.
- Report 4 transfer over interrupt endpoint from device to host to send data in response to SDP command in report 1. Max size of this report is 65 with first byte reserved for Report ID set to 4.

19.8.1.1 USB Configuration Details

The USB OTG function device driver supports a high speed (HS for UTMI) non-stream mode with a maximal packet size of 512 B and a low-level USB OTG function.

The VID/PID and strings for USB device driver are listed in the table below.

Table 19-52. VID/PID and Strings for USB Device Driver

Descriptor	Value
VID	0x15A2 (Freescale Vendor ID)
PID[1]	0x006A
String Descriptor1 (manufacturer)	Freescale Semiconductor Inc.
String Descriptor2 (product)	
String Descriptor4	Freescale Flash
String Descriptor5	Freescale Flash

1. Allocation based on BPN (Before Part Number)

19.8.1.2 IOMUX Configuration for USB

The interface signals of the UTMI PHY are not configured in the IOMUX. The UTMI PHY interface uses dedicated contacts on the IC. See the device data sheet for details.

19.8.2 UART Boot Flow

UART support in the device is composed of the UART controller(called as Host). The device ROM supports UART0, UART1, UART2 or UART3 ports for serial download.

Initially, the selected UART port for booting waits for a data pattern (0x23454523) from the UART host. After receiving this pattern, the device sends the same pattern back to the host. This phase of of pattern exchange is called Association phase.

Once the association phase is over, the host and device UART device implements SDP protocol of data transfers:

- 16 bytes SDP Commands are sent by the UART host
- Optionally more data bytes are sent by the UART host to the device UART device
- The device UART port sends 4 bytes of HAB security state, which is 0x12343412 in production mode and 0x56787856 in engineering mode.

If required, the device UART device sends data to UART host in response to the certain SDP commands.

19.8.2.1 UART eFUSE Configuration

The boot ROM determines the configuration of UART, either provided by eFUSE, or sampled on GPIO pins, during boot. See the table below for parameters details.

Table 19-53. UART eFUSE Descriptions

Fuse	Config	Definition	GPIO[1]	Shipped Value	Settings
BOOT_CFG3[5:4]	OEM	UART Boot Device Selection	Yes	00	00 – UART0 01 – UART1 10 – UART2 11 – UART3

1. Setting can be overridden by GPIO settings when BT_FUSE_SEL fuse is intact. See Table 19-3 for corresponding GPIO pin.

19.8.2.2 IOMUX Configuration for UART

The table below shows the UART IOMUX pin configuration.

Table 19-54. UART IOMUX Pin Configuration

Signal	Pad Name			
	UART0	UART1	UART2	UART3
TX	SCI0_TX	SAI0_TX_BCLK	QSPI0_A_SCK	TRACED[14]
RX	SCI0_RX	SAI0_RX_BCLK	QSPI0_A_CS0	TRACED[15]
RTS	SCI0_RTS	SAI0_RX_DATA	QSPI0_A_DATA[3]	SAI0_TX_BCLK
CTS	SCI0_CTS	SAI0_TX_DATA	QSPI0_A_DATA[2]	SAI0_RX_BCLK

19.8.3 Serial Download protocol

The 16 byte SDP command from host to device is sent using HID report 1. The table below describes 16 byte SDP command data structure:

Table 19-55. 16 Byte SDP Command Structure

BYTE Offset	Size	Name	Description
0	2	COMMAND TYPE	Following commands are supported for the device ROM: • 0x0101 READ_REGISTER • 0x0202 WRITE_REGISTER • 0x0404 WRITE_FILE • 0x0505 ERROR_STATUS • 0x0A0A DCD_WRITE • 0x0B0B JUMP_ADDRESS
2	4	ADDRESS	Only relevant for following commands: READ_REGISTER, WRITE_REGISTER, WRITE_FILE, DCD_WRITE, and JUMP_ADDRESS. For READ_REGISTER and WRITE_REGISTER commands, this field is address to a register. For WRITE_FILE and JUMP_ADDRESS commands, this field is an address to internal or external memory address.
6	1	FORMAT	Format of access, 0x8 for 8-bit access, 0x10 for 16-bit and 0x20 for 32-bit access. Only relevant for READ_REGISTER and WRITE_REGISTER commands.
7	4	DATA COUNT	Size of data to read or write. Only relevant for WRITE_FILE, READ_REGISTER, WRITE_REGISTER and DCD_WRITE command. For WRITE_FILE and DCD_WRITE commands DATA COUNT is in byte units.

Table continues on the next page...

Table 19-55. 16 Byte SDP Command Structure (continued)

BYTE Offset	Size	Name	Description
11	4	DATA	Value to write. Only relevant for WRITE_REGISTER command.
15	1	RESERVED	Not used in the device ROM

19.8.3.1 SDP Command

SDP commands are described in the following sections.

19.8.3.1.1 READ REGISTER

The transaction for command READ_REGISTER consists of following: 16 bytes (carried through Report1 in case of USB OTG interface) for command, HAB mode (carried through Report3 in case of USB OTG interface) and response or register value (carried through Report 4 in case of USB OTG interface). The register to read is specified in ADDRESS field of SDP command. First device sends HAB mode (through report-3 in case of USB OTG) followed by data bytes (carried through report-3 in case of USB OTG interface) read at given address. In case of USB OTG interface in use, if count is greater than 64 then multiple reports with report id 4 are sent until entire data requested by host is sent, while in case of other interfaces, the whole data is sent continuously The STATUS is either 0x12343412 for production parts and 0x56787856 for development parts.

Command, Host to Device (carried through Report1 in case of USB-OTG):

Report ID(For USB Only)	16 Bytes SDP Command
1	Valid values for READ_REGISTER COMMAND, ADDRESS, FORMAT, DATA_COUNT

Response, Device to Host:

	Response, Device to Host
3	4 bytes HAB mode indicating Production/Development part

Response, Device to Host: first response report

Report ID(For USB Only)	Response, Device to Host: first response
4	Register Value

If number of bytes requested is less than 4 then remaining bytes should be ignored by host.

In case of USB OTG only, multiple reports of Report ID 4 are sent until entire data requested is sent

Report4, Response, Device to Host: Last response report

Report ID(For USB Only)	Response, Device to Host: Last response report
4	Register Value

If number of bytes requested is less than 64 then remaining bytes should be ignored by host.

19.8.3.1.2 WRITE REGISTER

The transaction for command WRITE_REGISTER consists of following: 16 bytes (carried through Report1 in case of USB OTG interface) for command, HAB mode (carried through Report3 in case of USB OTG interface) and write status (carried through Report4 in case of USB OTG interface). Host sends WRITE_REGISTER command (using Report1 in case of USB OTG interface). The register to write is specified in ADDRESS field of SDP command, with FORMAT field set to data type (number of bits to write 8, 16 or 32) and value to write in DATA field of SDP command. Device writes the DATA to register address and returns WRITE_COMPLETE response code (using Report4 in case of USB OTG interface) and Development/Engineering part status (using Report3 in case of USB OTG interface) to complete the transaction.

Command, Host to Device:

Report ID(For USB only)	Command, Host to Device
1	Valid values for WRITE_REGISTER COMMAND, ADDRESS, FORMAT, DATA_COUNT and DATA

Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
3	4 bytes HAB mode indicating Production/Development part

Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
4	WRITE_COMPLETE (0x128A8A12) status

On failure device will report HAB error status.

19.8.3.1.3 WRITE_FILE

The transaction for command WRITE_FILE consists of following reports: 16 bytes (carried through Report1 in case of USB OTG interface) for command, HAB mode (carried through Report3 in case of USB OTG interface) and write status (carried through Report4 in case of USB OTG interface). Host sends WRITE_FILE command (through Report1 in case of USB OTG interface). The address to write is specified in ADDRESS field of SDP command (of Report1 in case of USB OTG interface), with FORMAT field set to data type (number of bits to write 8, 16, or 32). The data to be written to the address specified in SDP ADDRESS field then follows, from Host to Device (through Report2 in case of USB OTG interface). Device then returns COMPLETE code (using Report4 in case of USB OTG interface) and Development/Engineering part status (using Report3 in case of USB OTG interface) to complete the transaction:

Report1, Command, Host to Device:

Report ID(For USB only)	Command, Host to Device
1	Valid values for WRITE_FILE COMMAND, ADDRESS, DATA_COUNT

=======================Optional Begin=================

Host sends ERROR_STATUS command to query if HAB rejected the address

======================= Optional End==================

Data, Host to Device:

Report ID(For USB Only)	Data, Host to Device
2	File Data

In case of USB OTG interface, Data will be sent through Max 1024 bytes data per report. In case of DATA_COUNT more than 1024 bytes, host sends more Report2, Data, Host to Device:

Report2, Data, Host to Device:

Report ID(For USB only)	Data, Host to Device
2	File Data

Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
3	4 bytes HAB mode indicating Production/Development part

Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
4	COMPLETE (0x88888888) status

On failure device will report HAB error status.

19.8.3.1.4 ERROR_STATUS

The transaction for SDP command ERROR_STATUS consists of three parts. Host sends the command (through Report1 in case of USB OTG interface); device sends global error status in 4 bytes (using Report4 in case of USB OTG interface) after returning Development/Engineering status code (through Report3 in case of USB OTG interface). When device receives ERROR_STATUS command it will return global error status that is updated for each command. This command is useful to find out if last command resulted in device error or succeeded.

Command, Host to Device:

Report ID(For USB only)	Command, Host to Device
1	ERROR_STATUS COMMAND

Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
3	4 bytes HAB mode indicating Production/Development part

Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
4	4 bytes Error Status

19.8.3.1.5 DCD WRITE

The SDP command DCD_WRITE is used by host to send multiple register writes together. This command is provided to speed up the process of programming register writes such as to configure external RAM device. The command (sent through Report1 in case of USB OTG interface) goes with COMMAND TYPE set to DCD_WRITE, ADDRESS which is used for temporary location of DCD data and DATA_COUNT to number of bytes sent in data out phase. In data phase, host sends data for number of registers (using Report2 in case of USB OTG interface). Device completes the transaction (with Report3 in case of USB OTG interface) indicating Development/ Production STATUS and WRITE_COMPLETE (with report 4 in case of USB OTG interface) code 0x12828212.

Command, Host to Device:

Report ID(For USB only)	Command, Host to Device
1	DCD_WRITE COMMAND, ADDRESS, DATA_COUNT

Data, Host to Device:

Report ID(For USB only)	Datam Host to Device
2	DCD binary data

Max 1024 bytes per report in case of USB OTG only.

Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
3	4 bytes HAB mode indicating Production/Development part

Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
4	WRITE_COMPLETE (0x128A8A12) status

On failure device will report HAB error status.

See Device Configuration Data (DCD) for DCD format description.

19.8.3.1.6 JUMP ADDRESS

The SDP command JUMP_ADDRESS will be the last command host can send to the device, after this command device will jump to the address specified in the ADDRESS field of SDP command and start executing. This command should typically follow after WRITE_FILE command. The command is sent by host in command-phase of transaction (using Report1 in case of USB OTG interface), there is no data phase for this command but device send status (through report3 in case of USB OTG interface) to complete the transaction. And if HAB authentication fails then it will also send HAB error status (through report 4 in case of USB OTG interface).

Command, Host to Device:

Report ID(For USB only)	Command, Host to Device
1	JUMP_ADDRESS COMMAND

Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
3	4 bytes HAB mode indicating Production/Development part

Device sends a response only in case of an error jumping to the given address; device reports error with Response, Device to Host:

Report ID(For USB only)	Response, Device to Host
4	4 bytes HAB Error Status

19.9 Recovery Devices

The device ROM supports recovery devices. If primary boot device fails, boot ROM will try to boot from recovery device using one of I2C or SPI ports.

For enabling recovery device BOOT_CFG4 [6] fuse must be set. Additionally Serial EEPROM fuses must be set as described in Serial ROM Boot using SPI/I2C Interface.

19.10 HAB Re-Authentication at Low Power Standby Exit

This feature allows Boot code to be re-authenticated when silicon wakes up after entering low power standby. This Feature is controlled by BOOT_CFG3[0]. When fuse is set then this feature is enabled. During development stage this bit can be controlled using RCON16 (when BOOT_MODE[1:0]=0b10). When BT_FUSE_SEL efuse is blown, then the configuration setting is read from eFUSE. When enabled, the boot code is downloaded again from the boot interface (either Primary or Secondary) based on settings of PERSIST_SECONDARY_BOOT. After downloading the code is then re-authenticated using HAB based on settings of SEC_CONFIG[1:0] eFuse. Authenticated code is then allowed to execute. In case the authentication fails, the Boot ROM tries to download the image from the Recovery Device (depending on settings of BOOT_CFG4[6]). This feature is targeted to prevent execution of any malicious code or accidental modification of boot code that was introduced when the system was under STANDBY state.

19.11 Running Secondary Core

To start execution in the secondary (Cortex-M4) core, an application entry address must be established and the auxiliary core clock must be enabled. The auxiliary core begins execution in the Boot ROM code. The Boot ROM code checks the validity of the application entry address address pointed to by SRC_GPR2 and, if valid, jumps to the address.

1. Set SRC_GPR2[PERSISTENT_ENTRY1] to the entry point for the application to be executed by the secondary core. Ensure that this address is odd because the Cortex-M4 supports only Thumb Mode. For example,

   ```
   SRC.GPR2[PERSISTENT_ENTRY1] = 0x3f040410+1;
   ```

2. Set SRC_GPR3[PERSISTENT_ARG1] to the argument to be passed to the application.
3. Enable the auxiliary core clock by setting CCM_CCOWR to the key number 0x15a5a.

   ```
   CCM_CCOWR = 0x15a5a;
   ```

19.12 Appendix

19.12.1 IOMUX and GPIO Pad Settings for BOOT Interfaces

To view the IOMUX and GPIO Pad settings of Boot interfaces, refer to the BOOT_IOMUX_GPIO_SETTINGS excel file attached to this document. Locate the paperclip symbol on the left side of the PDF window, and click it. Double-click on the excel file to open it.

19.12.2 Fuse RCON Mapping

To view the RCON pins and Port pin mapping of BOOT interfaces, refer to Fuse_RCON_Mapping excel file attached to this document. Locate the paperclip symbol on the left side of the PDF window, and click it. Double-click on the excel file to open it.

19.12.3 PLL Configuration after BOOT

Table 19-56. PLL Configuration After Boot

PLL	Main		Fractional Divider 1		Fractional Divider 2		Fractional Divider 3		Fractional Divider 4	
	Mul	Freq (MHz)	Div	Freq (MHz)	Div	Freq (MHz)	Div	Freq (MHz)	Div	Freq (MHz)
PLL1	22	528	19	500.21	21	452.57	24	396	18	528
PLL2	22	528	19	500.21	24	396	28	339.42	23	413.21
PLL3	20	480	28	308.57	26	332.3	29	297.93	29	297.93

NOTE

PLL2 will be disabled after boot. PLLSYS and PLL3 are enabled during boot.

19.12.4 Basic CCM Settings

Table 19-57. CCM_Settings

Register	Value	Description
CCR	0x0001_10FF	FIRC_EN = 1, COSC_EN = 1, OSCNT = 0xFF
CSR	0x0000_0030	COSC_READY = 1
CCSR	0x0004_0824	PLL1_PFD_CLK_SEL = 0x4, PLL1_PFD4_EN = 1, FAST_CLK_SEL = 1, SYS_CLK_SEL = 0x4
CACRR	0x0060_0809	FLEX_CLK_DIV = 1, PLL6_CLK_DIV = 1, IPG_CLK_DIV = 1, BUS_CLK_DIV = 1, ARM_CLK_DIV = 1
CIMR	0xFFFF_FFFF	M_COSC_READY = 1, M_LRF_PLL4 = 1, M_LRF_PLL3 = 1, M_LRF_PLL2 = 1, M_LRF_PLL1 = 1

Chapter 20
Debug Architecture

20.1 Overview

This document describes the hardware and software debug and application development features and resources of this device.

The device debug and trace is based on ARM CoreSight architecture supplemented with the Secured JTAG controller (SJC) to allow security features.

1. Supports IEEE 1149.1 JTAG, IEEE 1149.7 JTAG, and ARM Serial Wire Debug (SWD) interface.
2. A range of security levels from NO JTAG to fully Open based on eFuse configuration for both debug and Test.
3. Support for Secured and non secured invasive/ non invasive debug to allow further granularity in debug accesses.
4. Support for field return parts to open access for debug and test to allow failure analysis.
5. Cross Trigger supported between the two cores as recommended by ARM.
6. Following debug capabilities are supported:
    a. Access to core and memory mapped resource examination and modification.
    b. Support for monitor mode and halt mode.
    c. Breakpoint/ Watchpoint control.
7. System profiling and performance monitoring.
8. CoreSight Embedded Trace Macrocell (ETM) and Instrumentation Trace Macrocell (ITM) to generate traces from the Cortex A5 and Cortex M4 cores.

9. Support ARM Real Time trace Interfaces: Trace Port Interface Unit (TPIU) and Serial Wire Output (SWO).
10. For TPIU trace port the maximum trace packets extraction bandwidth is limited by the pads and is equal to 160MByte/s (80Mhz x 16-bit).
11. For SWO the maximum frequency of operation is targeted for 24Mhz.

20.2 System Level Debug Architecture

This section discusses the overall debug and trace architecture from system perspective. The figure below gives the overall block diagram and design topology of the debug architecture.

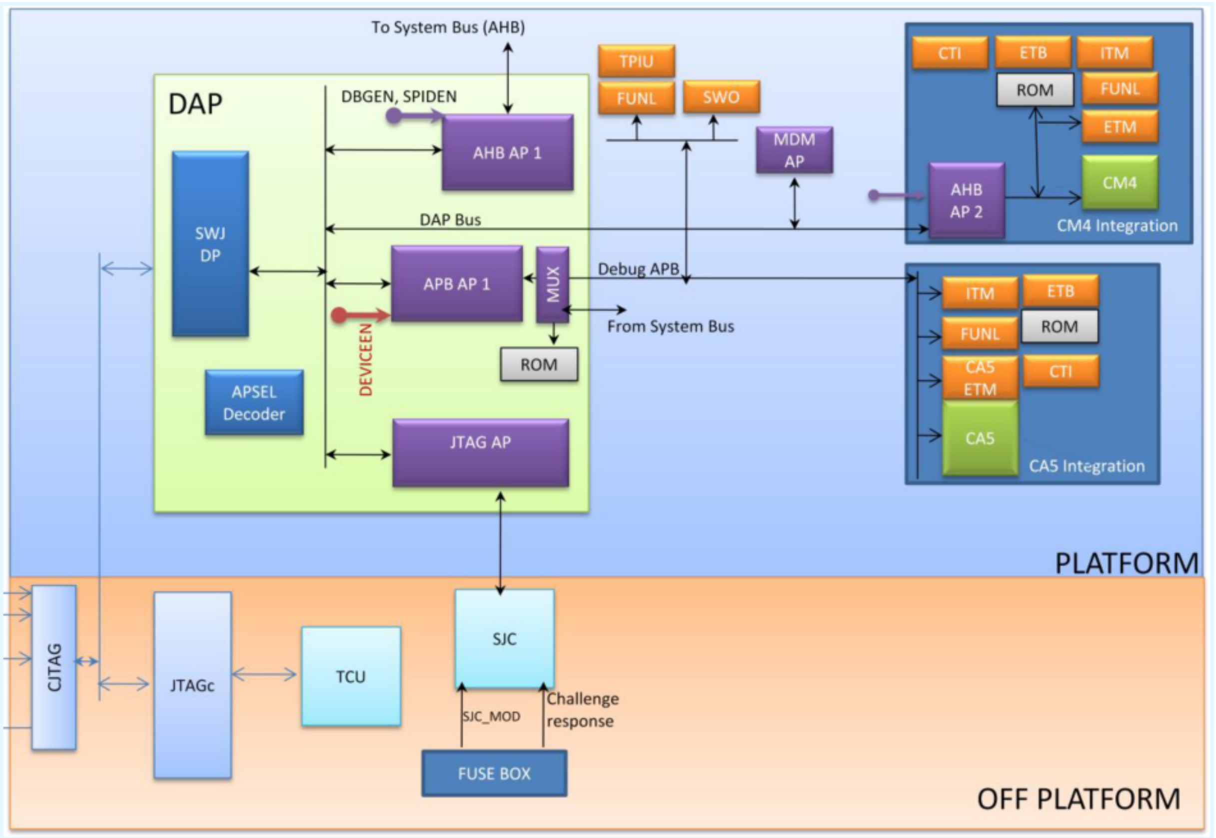


Figure 20-1. Debug Architecture Block Diagram

All ARM specific debug logic – Debug Access Port (DAP) and trace modules are integrated in the platform. The term platform is just a design partition consisting of cores integration, interconnect, debug, DMA, OCRAM controllers and a few other blocks.

The access to core and memory mapped resources is based on DAP. The access is controlled by security level configuration and authentication using the (Secure JTAG Controller) SJC block. SJC is accessible through the JTAG Access Port (JTAG AP) of DAP.

If the device is in fab or is a field return (both conditions set by way of eFuse configuration) all accesses to core/ memory mapped resources are fully available else these accesses are subject to authentication using SJC.

The TCU is fully accessible in fab and for field return; else it is completely blocked for security reasons.

The JTAGc is *always* available to allow boundary scan.

The debug IDCODE in case of JTAG DP is 0x4BA00477 and in case of SW DP it is 0x3BA02477.

20.3 Test and Debug Access Port Connectivity

The figure below gives a detailed TAP connectivity diagram.

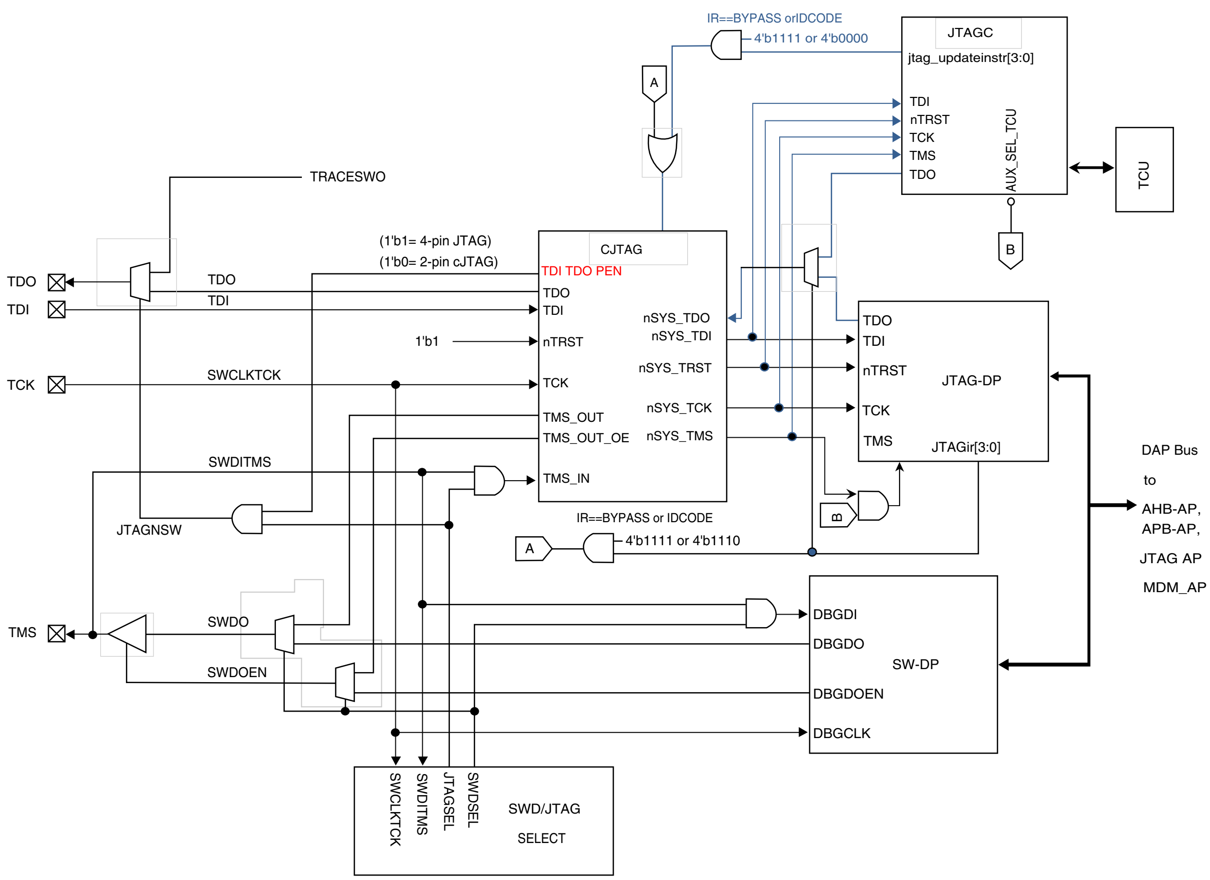


Figure 20-2. Detailed TAP connectivity diagram

In Figure 20-2 the *SWD/JTAG SELECT, SW-DP* and *JTAG-DP* are part of *ARM DAP*.

Please note the following points from Figure 20-2 :

1. The Debug port comes out of reset in a standard 4 wire JTAG mode only. It is switched into SWD or cJTAG mode by change sequences described in JTAG-to-SWD change sequence and JTAG-to-cJTAG change sequence. Once the mode has been changed, unused debug pins can be reassigned to any of their alternative muxed functions. Once switched to C-JTAG or SWD, switching back to 4-Wire JTAG is not possible.
2. The JTAG pads are connected to cJTAG as the primary interface. Besides the TMS is also observed by the SWD/ JTAG switcher logic of the DAP (*SWDITMS* input in *SWD/JTAG SELECT block*).
3. The JTAG DP and JTAGC are connected to cJTAG in an *overlay scheme*; both have an IR length of *four*.
4. The TCU is an auxiliary TAP on the JTAGC.
5. If TCU is selected *(AUX_SEL_TCU goes HIGH),* the TMS input for the JTAG-DP is driven LOW (Refer to connector "B" in the diagram). Thus giving exclusive access to the auxiliary TAP (this allows auxiliary TAP to have an IR length greater than four).
6. If the DAP switches from JTAG to SWD (when SWD/ JTAG SELECT receives the switching sequence), the TMSIN for cJTAG is driven low (as JTAGSEL goes LOW) and the TMS pad is used for SWD data inout.

20.4 JTAG to SWD cJTAG switching sequence

20.4.1 JTAG-to-SWD change sequence

- Send more than 50 TCK cycles with TMS(SWDIO) =1
- Send the 16-bit sequence on TMS(SWDIO) = 0111_1001_1110_0111 (MSB transmitted first)
- Send more than 50 TCK cycles with TMS (SWDIO) =1.

  This is shown in the Figure 20-3.

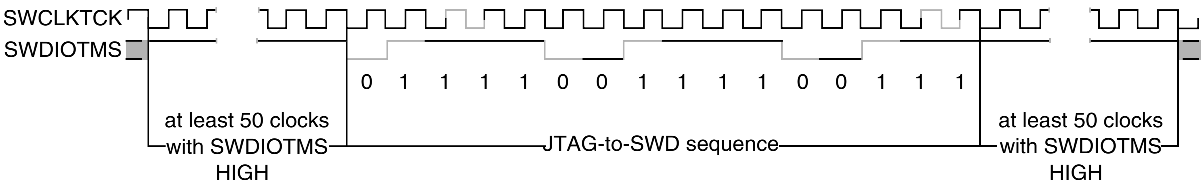


Figure 20-3. JTAG-to-SWD switching sequence

20.4.2 JTAG-to-cJTAG change sequence

- Reset the debug port.
- Set the control level to 2 via the zero-bit scans.
- Execute the Store Format (STFMT) command (00011) to set the scan format register to 1149.7 scan format.

20.4.3 System JTAG Controller (JTAGC)

The system JTAG controller (JTAGC) is connected in parallel with ARM TAP controller (JTAG DP of ARM DAP). The IR length is 4-bits. The JTAGC IR codes overlay the ARM DAP controller IR codes. JTAGC uses *twelve* instructions and the DAP uses rest *four*. The outputs of the TAPs (TDO) are muxed based on the IR Code which is selected. This design is fully JTAG compliant and appears to the JTAG chain as a single TAP.

The Table 20-1 gives the IR codes for the system JTAG controller. For the instructions used by ARM DAP TAP refer to Table 20-2.

Table 20-1. JTAG instructions for system JTAG controller (JTAGc)

Code	JTAGC IR
4'b0000	IDCODE
4'b0001	CENSOR_CTRL
4'b0010	SAMPLE-PRELOAD
4'b0011	SAMPLE
4'b0100	EXTEST
4'b0101	HI-Z
4'b0110	TEST_CTRL
4'b0111	TEST_LEAKAGE
4'b1000	*Not used by JTAGC (used by DAP)*
4'b1001	AUX ACCESS 1 (TCU selection)
4'b1010	*Not used by JTAGC (used by DAP)*

Table continues on the next page...

Table 20-1. JTAG instructions for system JTAG controller (JTAGc) (continued)

Code	JTAGC IR
4'b1011	*Not used by JTAGC (used by DAP)*
4'b1100	Reserved
4'b1101	CLAMP
4'b1110	*Not used by JTAGC (used by DAP)*
4'b1111	BYPASS

The Test Control Unit (TCU) is an auxiliary TAP on the JTAGc. It is selected by the instruction "AUX ACCESS 1". When TCU is selected the DAP JTAG DP is disconnected (TMS input to JTAG DP goes low), giving TCU exclusive access of the JTAG port. Refer to id-_Ref302467261. DFT Specification for details related to TCU.

20.4.4 Debug Access Port (DAP) TAP

DAP is a standard ARM component. The DAP provides multiple master driving ports, all accessible and controlled through a single external interface port to provide system-wide debug.

As discussed earlier the DAP IR overlay with the system JTAG controller IR. The DAP Instruction Registers is listed in Table 20-2.

DAP uses five Instructions but since BYPASS is identical with JTAGC it has been dropped out.

Table 20-2. ARM DAP IR Codes

Code	DAP IR
4'b1000	ABORT
4'b1010	DPACC
4'b1011	APACC
4'b1110	IDCODE

The DAP offers AHB and APB master interfaces to access system buses. It also exports the internal DAP bus to allow to extend the access ports as per the system requirement. It offers JTAG AP to allow adding auxiliary TAPs. The device uses JTAG AP to connect the Secure JTAG controller (SJC) block as an auxiliary TAP. For more details on SJC refer to id-_Ref302670360. For more information on DAP TAP refer to ARM Debug Interface v5 Architecture specification [id-_Ref302468262].

The AHB AP provides the debugger access to all memory and registers in the system. System access is independent of the processor status. AHB-AP does not do back to back transactions on the bus, so all transactions are non sequential.

The APB AP is used to access the CoreSight components for the CA5 and also the shared debug components of the system trace like TPIU, SWO.

The exported DAP bus is used to host AHB AP access port (integrated as part CM4 integration) and the Miscellaneous Debug module Access Port (MDM AP).

The MDM AP hosts system level JTAG status and control registers (refer to Debug Status and Control Registers) which can be used for cross triggering, synchronized debug, low power and other misc control/ status.

The selection of different access ports in the DAP is done based on the APSEL value set in the SELECT register of SWJ-DP. APSEL is 31:24 bit field of the DAP SELECT register.

The APSEL will be decoded as given in Table 20-3.

Table 20-3. APSEL decode

APSEL (SELECT[31:24])	Selection	Identification Register (IDR) value
8'h00	AHB-AP to System Bus (NIC-301)	0x4477_0001
8'h01	APB-AP to "Shared" debug IP and CA5 debug IP	0x2477_0002
8'h02	JTAG-AP to SJC	0x1476_0010
8'h03	AHB-AP to CM4	0x2477_0011
8'h04	Miscellaneous Debug module AP (MDM-AP)	0x001C_0000
8'hø5- 8'hff	Reserved (Default AP response)	-

20.4.4.1 ROM Table

The ROM table is used to hold the information about the debug components.

There are three ROM tables.

1. The CM4 ROM table resides on the AHB AP2 and has entries for CM4 debug components.

2. The DAP ROM table resides on the APB AP. It has entries for the common trace components, MDM AP and the last entry points to the CA5 ROM table. The DAP ROM table last entry has a "valid" bit (bit-0 in the ROM table entry). When this bit is "0", the last entry returns 0x00000000. When this bit is "1" the last entry returns the pointer to CA5 ROM table.
3. The CA5 ROM Table resides on APB AP. This has entries for CA5 debug components.

Table 20-4. ROM Table Entries bit assignment

Bits	Name	Description
[31:12]	Address offset	Base address of the component, relative to the ROM address. Negative values are permitted using two's complement. ComponentAddress = ROMAddress + (AddressOffset SHL 12).
[11:2]	-	Reserved SBZ.
[1]	Format	1 = 32-bit format. In the DAP Debug ROM this is set to 1. 0 = 8-bit format.
[0]	Entry present	Set HIGH to indicate an entry is present.

The device supports three core configurations:

1. Both CA5 and CM4 are present: All ROM tables are accessible and "valid" bit is set.
2. CM4 only: CM4 and DAP ROM tables are accessible. DAP ROM table "valid" bit is cleared and hence CA5 ROM table is not accessible.
3. CA5 Only: DAP ROM table is accessible and the "valid" bit is set, so CA5 ROM table is accessible. In this case the CM4 ROM table is not accessible.

   Note for Implementation: In case of "CA5 only" the clock for the CM4 should be gated and the DBGEN/ SPIDEN for the AHB AP2 in CM4 integration should be held LOW.

20.4.4.1.1 CM4 ROM table

COMPONENT	ADDRESS	Value
NVIC	0xE00FF000	0xFFF0F003
DWT	0xE00FF004	0xFFF02003
FPB	0xE00FF008	0xFFF03003
ITM	0xE00FF00C	0xFFF01003
TPIU	0xE00FF010	0xFFF41002

Table continues on the next page...

COMPONENT	ADDRESS	Value
ETM	0xE00FF014	0xFFF42003
ETB	0xE00FF018	0xFFF43003
FUNNEL	0xE00FF01C	0xFFF44003
CTI	0xE00FF020	0xFFF45003
END MARKER	0xE00FF024	0x00000000
MEMTYPE	0xE00FFFCC	0x00000001
PID4	0xE00FFFD0	0x00000004
PID5	0xE00FFFD4	0x00000000
PID6	0xE00FFFD8	0x00000000
PID7	0xE00FFFDC	0x00000000
PID0	0xE00FFFE0	0x000000C4
PID1	0xE00FFFE4	0x000000B4
PID2	0xE00FFFE8	0x0000000B
PID3	0xE00FFFEC	0x00000000
CID0	0xE00FFFF0	0x0000000D
CID1	0xE00FFFF4	0x00000010
CID2	0xE00FFFF8	0x00000005
CID3	0xE00FFFFC	0x000000B1

20.4.4.1.2 CA5 ROM table

Component	Address	Value
CA5-DBG	0x4008C000	0xffffC003
CA5-PMU	0x4008C004	0xffffD003
	0x4008C008	0xffffE002
	0x4008C00C	0xffffF002
	0x4008C010	0x00000002
	0x4008C014	0x00001002
	0x4008C018	0x00002002
	0x4008C01C	0x00003002
CA5-CTI	0x4008C020	0x00002003
	0x4008C024	0x00003002
	0x4008C028	0x00004002
	0x4008C02C	0x00005002
CA5-ETM	0x4008C030	0xffffE003
	0x4008C034	0xffffF002
	0x4008C038	0x00000002
	0x4008C03C	0x00001002
	0x4008C040	0x00000000

Table continues on the next page...

Component	Address	Value
	0x4008CFCC	0x00000000
PID4	0x4008CFD0	0x00000004
PID5	0x4008CFD4	0x00000000
PID6	0x4008CFD8	0x00000000
PID7	0x4008CFDC	0x00000000
PID0	0x4008CFE0	0x000000a5
PID1	0x4008CFE4	0x000000b4
PID2	0x4008CFE8	0x0000000b
PID3	0x4008CFEC	0x00000000
CID0	0x4008CFF0	0x0000000D
CID1	0x4008CFF4	0x00000010
CID2	0x4008CFF8	0x00000005
CID3	0x4008CFFC	0x000000B1

20.4.4.1.3 DAP ROM table

Component	Address	Value	
		CM4 only	**CM4+ CA5**
CA5-ETB	0x40087000	0x00000000	0x0000A003
No component	0x40087004	0x00000002	0x00000002
pltf-TPIU	0x40087008	0x0000D003	0x0000D003
pltf-FUNNEL	0x4008700C	0x0000E003	0x0000E003
CA5-ITM	0x40087010	0x00000000	0x00009003
pltf-SWO	0x40087014	0x0000F003	0x0000F003
CA5-FUNNEL	0x40087018	0x00000000	0x0000B003
CA5-ROM	0x4008701C	0x00000000	0x00005003
END MARKER	0x40087020	0x00000000	
Reserved	0x40087FCC	0x00000000	
PID4	0x40087FD0	0x00000000	
PID5	0x40087FD4	0x00000000	
PID6	0x40087FD8	0x00000000	
PID7	0x40087FDC	0x00000000	
PID0	0x40087FE0	0x00000080	
PID1	0x40087FE4	0x000000E9	
PID2	0x40087FE8	0x00000008	
PID3	0x40087FEC	0x00000001	
CID0	0x40087FF0	0x0000000D	
CID1	0x40087FF4	0x00000010	
CID2	0x40087FF8	0x00000005	
CID3	0x40087FFC	0x000000B1	

20.5 Debug Port Pin Descriptions

The debug port pins default after reset to their JTAG functionality and can be later reassigned to their alternate functionalities. In cJTAG and SWD modes TDI can be configured to alternate GPIO functions. Refer to Table 20-5 for pin assignments in different modes.

Table 20-5. Debug Port Pins

Pin Name	JTAG Debug Port		cJTAG Debug Port		SWD Debug Port		Internal Pullup/Down
	Type	Description	Type	Description	Type	Description	
TMS/SWDIO	I/O	JTAG Test mode selection	I/O	cJTAG Data	I/O	Serial Wire Data	Pull-up
TCK/SWCLK	I	JTAG Test clock	I	cJTAG Clock	I	Serial wire clock	Pull-down
TDI	I	JTAG Test Data input	-	-	-	-	Pull-up
TDO/TRACESWO	O	JTAG Test Data output	O	Trace output over a single pin	O	Trace output over a single pin	NC

20.6 Secure JTAG Controller (SJC)

SJC is accessed through the JTAG AP of the DAP. The SJC has a 5-bit Instruction Register which is described in Table 20-6.

Table 20-6. System JTAG Controller Instruction Registers

Code	SJC IR
5'b00000 - 5'b00011	Reserved
5'b00100	Enable Extra misc registers
5'b00101 - 5'b01011	Reserved
5'b01100	Security Output Challenge
5'b01101	Security input response
5'b01110 – 5'b11110	Reserved
5'b11111	BYPASS

The Extra misc registers IR allows accessing the extra miscellaneous registers. These include the register to understand the status of the authentication in case of challenge/ response. There can be other uses of these registers which the SOC can define.

The Security Output Challenge and Security Input Response IRs allows the debugger to read the security challenge and feed the response for authentication in case the Challenge-Response based authentication is in place. For details of the SJC block refer to System JTAG Controller (SJC) chapter.

The various security level configurations are discussed in section *Secured JTAG*.

20.6.1 Challenge Response Access Sequence

When JTAG_SMODE is 2'b01 (Secure JTAG mode is enabled), Fuse SJC_CHALL[63:0] and SJC_RESP[53:0] would have programmed. Now following steps need to follow to pass the challenge response.

1. Program SPIDEN and SPNIDEN in SRC_SECR_DOUT register through Software (if secured traces are required: this is possible only for authenticated software running in secured world).

2. PowerUp DAP

3. Select Port 0 for SJC in JTAG-AP PORT SELECT registers.

4. Reset SJC TAP by writing into JTAG-AP CSW register. Set TRST_OUT = 1. Also set TMS High for 5 clocks in BWFIFO 1 register change the bank select. After some random delay, release TRST_OUT.

5. Move SJC TAP to IDLE state.

6. Shift in the Challenge Instruction.

7. Read out the challenge Value.

8. Shift IR RESPONSE Instruction.

9. Shift DR Response 56 bit value.

10. READ MDM AP STATUS Register if Debug Enable bit is Set and Secure Debug bit is set

If the First attempt with Incorrect Challenge Response results into failure, subsequent attempts of correct Challenge Response will still be discarded. Only POR reset will let Debugger start the attempts from beginning.

20.7 Debug Status and Control Registers

The debugger has access to the status and control elements, implemented as registers in MDM-AP on the DAP bus as shown in Figure 20-1. These registers provide additional control and status for low power mode recovery and typical run-control scenarios. The status register bits also provide a means for the debugger to get updated status of the core without having to initiate a bus transaction across the crossbar switch, thus remaining less intrusive during a debug session.

It is important to note that these DAP control and status registers are not memory mapped within the system memory map and are only accessible via the Debug Access Port (DAP) using JTAG, cJTAG, or SWD. The MDM-AP is accessible as Debug Access Port 4 (refer to Table 20-3). The MDM-AP registers are listed in Table 20-7.

Table 20-7. MDM-AP Register Summary

Address	Register	Description
0x0400_0000	Status	MDM AP Status Register (Refer Table 20-9)
0x0400_0004	Control	MDM AP Control Register (Refer Table 20-8)
0x0400_00FC	ID	Read only Identification register that always reads as 0x001C_0000

20.7.1 Miscellaneous Debug Module (MDM) AP Control Register

The debugger can write to MDM-AP control register only if the DBGEN is SET ie the debugger security authentication has been cleared. In case the debugger makes an access while DBGEN=0, the access will be simply ignored and no error will be reported.

Table 20-8. MDM AP Control Register Assignments

Bit	Name	Description
0-1	Reserved	Not used
2	CM4 Debug Request	Drives "EDBGREQ" input for CM4. When the core goes into debug state it acknowledges with "HALTED" output signal. If the core is in STOP mode this bit is used to wake-up the core and transition to halted state.
3	System reset request (SYSRESETREQ)	Set to force the system reset. The system remains in reset till this bit remains asserted. When this bit is reset the entire system comes out of reset.
4	Reserved	unused

Table continues on the next page...

Table 20-8. MDM AP Control Register Assignments (continued)

Bit	Name	Description
5	Standby mode debug request (STANDBYDBGREQ)	Set to configure the system to be held in reset after the next recovery from standby modes (LPSTOP1/2/3). This bit is sampled by the SRC as "HOLD reset latch" just before the GPC moves the part in any of the LPSTOP modes and remains latched in always ON domain during the standby phase. On exit from standby mode SRC holds the CM4 and CA5 in reset at the end of system reset sequencing until hold reset release is asserted. The STANDBYDDBREQ is cleared due to POR reset generated as part of LPSTOP recovery.
6	Standby mode debug acknowledge (STANDBYDBGACK)	Set to release the cores being held in reset following a standby recovery due to SRC "HOLD reset latch". This bit drives directly the SRC to control this feature. Besides this bit also clears the "standby mode exit" status bit. The debugger re-initializes all debug IP and then asserts this control bit to allow the SRC to release the cores from reset and allow CPU operation to begin. The STANDBYDBGACK bit is cleared by JTAG or clears automatically due to POR reset generated as part of standby recovery.
7	Reserved	Unused
8	Timestamp disable	Freeze Time stamp during HALT mode. 0: The timestamp counter continues to count when the core is halted. 1: The timestamp counter freezes when either of the cores is halted. This allows to avoid generation of generation of timestamp overflow messages when the core is halted.
9	Inhibit Sleep	This bit is monitored by the power controller. If set the device will not move into power gated mode. Thus debugger can make safe accesses without risk of losing communication due to debug logic going into power down.
10-17	Reserved	Unused
18	CA5 Debug Request	Drives "EDBGREQ" input for CA5. When the core goes into debug state it acknowledges with "HALTED" output signal. If the core is in STOP mode this bit is used to wake-up the core and transition to halted state.
19-23	Reserved	unused
24	CTI trigger input	Connects to CTI trigger input
25	CTI trigger output ack Reserved	Acknowledgement for the CTI trigger out received on bit-22 of the MDM-AP status register. TKT085997 : fix postponed.
26-29	Reserved	unused
30	CM4 debug restart	CM4 debug restart input to CM4 core.
31	CA5 debug restart	CA5 debug restart input to CA5 core.

20.7.2 Miscellaneous Debug Module (MDM) AP Status Register

Table 20-9. MDM-AP Status Register Assignments

Bit	Name	Description
0-1	Reserved	Not used
2	System reset	Indicates the system reset state 1: system not in reset 0: system is in reset
3	debug access enable	Indicates if the debugger can access to AHB and APB AP. This bit reflects the status of DBGEN (which is same as NIDEN).
4	debug secured access enable	Indicates if the debugger can make secured accesses to the debug logic and can trace out secured traces. This bit reflects the status of SPIDEN/ SPNIDEN. SPIDEN and SPNIDEN are same and are controlled by configurable bit by secured software.
5-9	Reserved	Not used
10	Standby mode exit	This bit indicates an exit from standby mode has occurred. The debugger had lost communication while the system is in standby mode (including access to this register). Once communication is reestablished, this bit indicates that the system had been standby mode. Since the debug modules lose their state during standby mode, they need to be reconfigured. This bit is implemented in SoC glue logic and is set during the standby mode recovery sequence. The standby Mode Exit bit is held until the debugger has had a chance to recognize that a standby mode was exited and is cleared by a write of 1 to the STANDBYDBGACK bit[6] in MDM AP Control register.
11-15	Reserved	Not used
16	CM4 Halted	CM4 has entered debug halt mode
17	CM4 SLEEPDEEP	CM4 asserted SLEEPDEEP
18	CM4 SLEEPING	CM4 asserted SLEEPING
19	CA5 halted	CA5 has entered debug halt mode
20	CA5 STANDBYWFI	
21	CA5 STANDBYWFE	
22	CTI trigger out	Cross Trigger Interface Trigger out asserted.
23-29	Reserved	unused
30	CM4 debug restarted	Debug restarted output status from CM4.
31	CA5 debug restarted	Debug restarted output status from CA5.

Note for Implementation: The MDM-AP control register is accessible only after debug authentication. The MDM-AP status register is always accessible.

20.8 Debug Resets

The debug system receives the following sources of reset:

- Debug reset (CDBGRSTREQ bit within the SWJ-DP CTRL/STAT register) in the TCLK domain that allows the debugger to *reset the debug logic*. This includes ATRESETn for cm4_atbasync, csfunnel, csreplicator, csswo, cstpiu, CSATBUPSIZER_etm, CSATBUPSIZER_itm, CSREPLICATOR, CSETB, ca5_itm_upsizer) AND PRESETDBGn for csfunnel, csswo, cstpiu, DAP APs, faraday_dbg_trace_ctrl, CSTFUNNEL, CSETB, CSCTI, ca5_itm, ca5_funnel, ca5_etb.
- System POR reset.

Conversely the debug system is capable of generating system reset using the following mechanism:

- A system reset in the MDM-AP control register which allows the debugger to hold the *system* in reset.

20.9 Trace Architecture

The figure below shows the trace architecture for the device.

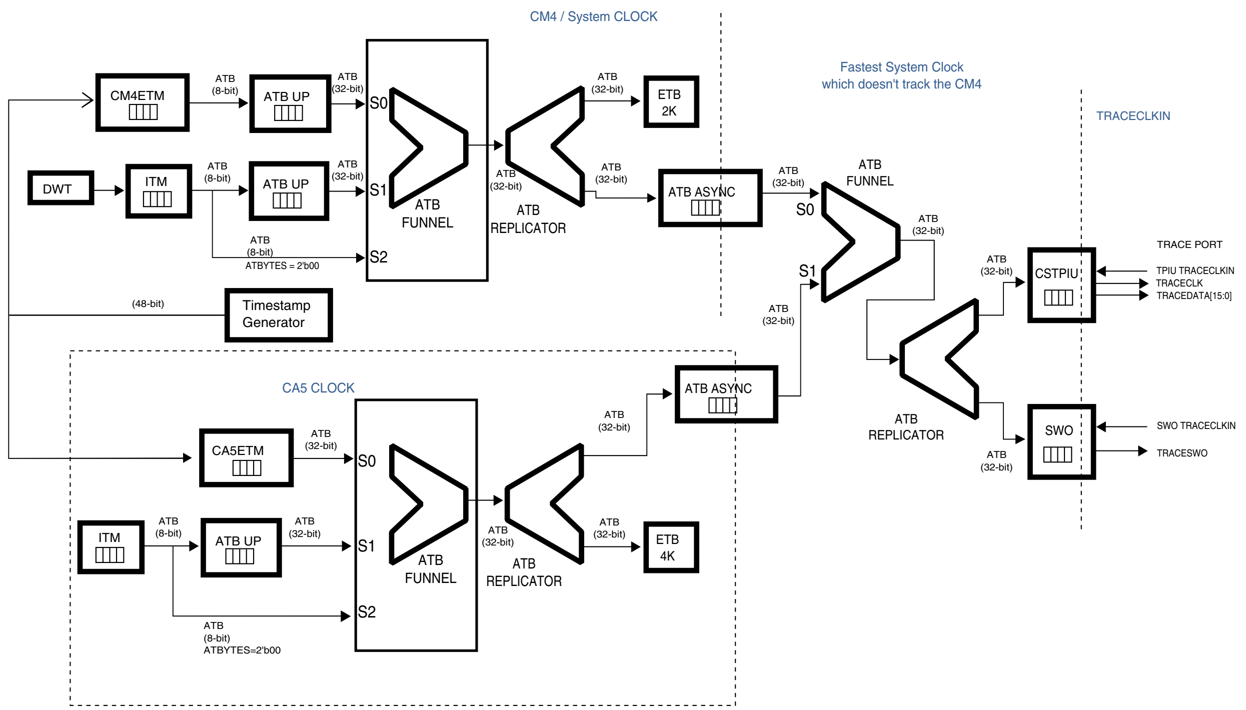


Figure 20-4. Trace Bus Connectivity diagram

The following Trace Paths are supported:

- Trace paths on a Cortex-M4 only device:

CM4 ATB path to TPIU enabled, CA5 ATB path to disabled to ensure that the CA5 ATB isn't requesting a transaction

- Trace paths on a Cortex-A5 only device:

CM4 ATB path to TPIU disabled, CA5 ATB path to enabled to ensure that the CM4 ATB isn't requesting a transaction.

- Trace paths on a dual-core device:

a) CM4 ETM/ITM to ETB and CA5 ETM/ITM to ETB. The TPIU is disabled; both trace path disable bits can be left enabled (or disabled).

b) CM4 ETM/ITM to TPIU and CA5 ETM/ITM to TPIU. Both trace paths (CM4 ATB Path and CA5 ATB path) enabled.

c) CM4 ETM/ITM to ETB and CA5 ETM/ITM to TPIU. The trace path from the CM4 to TPIU must be disabled.(CM4 ATB path to TPIU disabled, CA5 ATB path to enabled)

d) CM4 ETM/ITM to TPIU and CA5 ETM/ITM to ETB. The trace path from the CA5 to TPIU must be disabled. (CM4 ATB path to TPIU enabled, CA5 ATB path to disabled)

In case SWO is enabled as the output port, the possible trace paths are

a) CA5 ITM -> CA5 ATB FUNNELS2 -> CA5 ATB REPL -> ATB ASYNC -> shared ATB funnel -> shared ATB RELP -> SWO (in this path only LSByte of the ATB data bus is used)

b) CM4 ITM -> CM4 ATB FUNNELS2 -> CM4 ATB REPL -> ATB ASYNC -> shared ATB funnel -> shared ATB RELP -> SWO (in this path only LSByte of the ATB data bus is used)

c) We can have traces from both CA5 and CM4 ITMs active and these merged by shared ATB funnel and traced out to SWO.

d) One of path a) or b) active while the other core traces logged to the corresponding ETB.

Note

There are *four* clock domains in the trace connectivity. The trace components associated with the CM4 and CA5 work in respective core clock domains. These are separated from the common trace components at the system level through ATB Async interfaces. This allows us to clock the common

components at a fixed high frequency clock (limited by the pad frequency) thereby offering highest possible output bandwidth for draining out the traces, independent of the core frequencies. The TRACECLKIN is driven internally from a mux select from different sources and feeds both TPIU and SWO.

The Platform Trace Control (Pltf-TCTL) module controls the trace flow: The register is at the offset 0x4009_3000 with the following fields:

- Ctrl[0] = CM4 to TPIU Disable
- Ctrl[1] = CA5 to TPIU Disable
- Ctrl[2] = CM4ITM to SWO Enable
- Ctrl[3] = CM5ITM to SWO Enable

20.9.1 Data Watchpoint and Trace (DWT)

DWT is part of *Cortex M4 trace connectivity only*. The DWT is a unit that performs the following debug functionality:

- It contains four comparators that you can configure as a hardware watchpoint, an ETM trigger, a PC sampler event trigger, or a data address sampler event trigger. The first comparator, DWT_COMP0, can also compare against the clock cycle counter, CYCCNT. The second comparator, DWT_COMP1, can also be used as a data comparator.
- The DWT contains counters for:
  - Clock cycles (CYCCNT)
  - Folded instructions
  - Load store unit (LSU) operations
  - Sleep cycles
  - CPI (all instruction cycles except for the first cycle)
  - Interrupt overhead

20.9.2 Flash Patch and Breakpoints (FPB) (CM4 only)

The FPB implements hardware breakpoints and patches code and data from code space to system space. This is applicable to *CM4 core only*.

The FPB unit contains two literal comparators for matching against literal loads from Code space, and remapping to a corresponding area in System space.

The FBP also contains six instruction comparators for matching against instruction fetches from Code space, and remapping to a corresponding area in System space. Alternatively, the six instruction comparators can individually configure the comparators to return a Breakpoint Instruction (BKPT) to the processor core on a match, so providing hardware breakpoint capability.

20.9.3 Instrumentation Trace Macrocell (ITM)

The ITM is an application-driven trace source that supports printf style debugging to trace Operating System (OS) and application events, and emits diagnostic system information. The ITM emits trace information as packets. There are three sources that can generate packets. If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output.

The three sources in decreasing order of priority are:

1. Software trace -- Software can write directly to ITM stimulus registers. This emits packets.
2. Hardware trace -- The DWT generates these packets, and the ITM emits them (not supported by CA5 ITM)
3. Time stamping -- Timestamps are emitted relative to packets. The ITM contains a 21-bit counter to generate the timestamp. The Cortex-M4 clock or the bit clock rate of the Serial Wire Viewer (SWV) output clocks the counter.

20.9.4 Embedded Trace Macrocell (ETM)

The ETM v3.5 is supported. The Cortex-M4 Embedded Trace Macrocell (ETM-M4) is a debug component that enables a debugger to reconstruct program execution. The CoreSight ETM-M4 supports only instruction trace. You can use it either with the Cortex-M4 Trace Port Interface Unit (M4-TPIU), or with the CoreSight ETB.

The main features of an ETM are:

- tracing of 16-bit and 32-bit Thumb instructions
- four EmbeddedICE watchpoint inputs
- a Trace Start/Stop block with EmbeddedICE inputs

- one reduced function counter
- two external inputs
- a 24-byte FIFO queue (cortex M4).
- global timestamping

20.9.5 CoreSight Embedded Trace Buffer (ETB)

The ETB provides on-chip storage of trace data using 32-bit RAM. The ETB accepts trace data from any CoreSight-compliant component trace source with an ATB master port, such as a trace source or a trace funnel. It is included in this device to remove dependencies from the trace pin pad speed, and enable low cost trace solutions. The device has two ETBs – one each for CM4 and CA5 traces. The ETB for collecting CM4 traces is 2KB while the one to collect CA5 traces is 4KB.

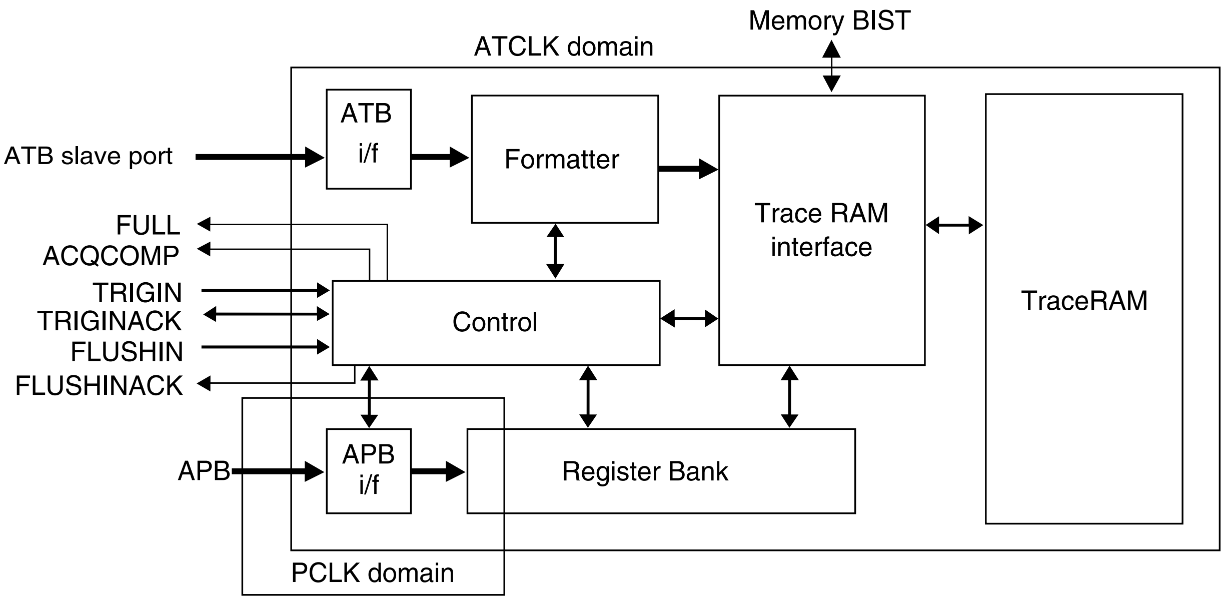


Figure 20-5. ETB Block Diagram

The ETB contains the following blocks:

- Formatter -- Inserts source ID signals into the data packet stream so that trace data can be re-associated with its trace source after the data is read back out of the ETB.
- Control -- Control registers for trace capture and flushing.
- APB interface -- Read, write, and data pointers provide access to ETB registers. In addition, the APB interface supports wait states through the use of a PREADYDBG signal output by the ETB. The APB interface is synchronous to the ATB domain.

- Register bank -- Contains the management, control, and status registers for triggers, flushing behavior, and external control.
- Trace RAM interface -- Controls reads and writes to the Trace RAM..

20.9.6 Trace Port Interface Unit (TPIU)

TPIU acts as a bridge between the on chip trace data from ETM and ITM, with separate IDs to a data stream encapsulating IDs where required to provide external visibility of the packet stream to a Trace Port analyzer. The device offers a synchronous trace port consisting of 16-bit TRACEDATA, TRACECTL and TRACECLK.

20.9.7 Serial Wire Output

The CoreSight SWO is a trace data drain that acts as a bridge between the on-chip trace data to a data stream that is captured by a *Trace Port Analyzer* (TPA). The device also offers an asynchronous Serial Wire out (TRACESWO) interface. The SWO can work at a maximum frequency of 24Mhz.

20.9.8 Performance Monitoring Unit (for CA5 Only)

Each core in the Cortex-A5 MPCore processor contains a PMU which provides two counters to gather statistics on the operation of the core and memory system. Each counter can count any of the events available in the Cortex-A5 MPCore processor. It also provides a single 32-bit cycle counter with support for scaling and filtering on the processor mode and security state. See the *ARM Architecture Reference Manual, ARMv7-A* and *ARM ArchitectureReference Manual Performance Monitors v2 Supplement* for more information about performance monitoring.

20.9.9 Embedded Cross Trigger

Embedded Cross Trigger (ECT) is used for multi core run control and trace cross triggering for example synchronous stop start for all cores or trigger program trace on a trigger event from another core or IP.

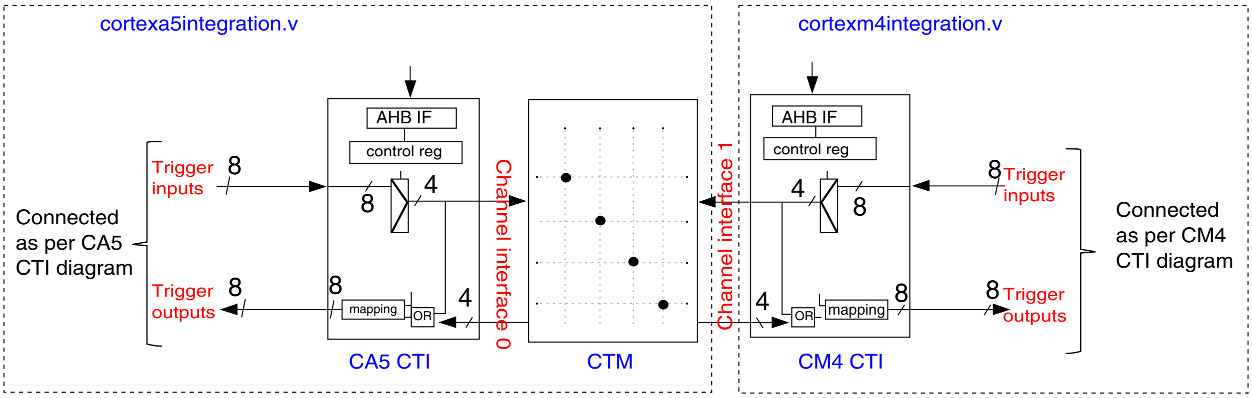
ECT is made up of Cross Trigger Interfaces (CTI) and Cross Trigger Matrix (CTM).

Cross Trigger Interface (CTI) provided by ARM:

- 8 trigger inputs, 8 trigger outputs, basic signal conditioning
- Programmable channel routing (mapping trigger inputs to channels and channels to trigger output.
- Limited to 4 channels

Cross Trigger Matrix (CTM) provided by ARM:

- Provides channel routing when more than 2 CTIs are used.
- Limited to 4 channels



Note: Channel Interface 2 and 3 are unused.

20.9.9.1 CM4 CTI Triggers

CM4 CTI Trigger Inputs:

Table 20-10. CM4 CTI Trigger inputs

Trigger Bit	Source Signal	Source Device
[7]	ETMTRIGOUT	ETM
[6]	ETMTRIGGER[2]	DWT
[5]	ETMTRIGGER[1]	DWT
[4]	ETMTRIGGER[0]	DWT
[3]	ACQCOMP	ETB
[2]	FULL	ETB
[1]	MDMAP CTITRIGIN (MDM-AP Control[24])	MDM-AP
[0]	DBGACK	Core

CM4 CTI Trigger Outputs:

Table 20-11. CM4 CTI Trigger outputs

Trigger Bit	Source Signal	Source Device
[7]	DBGRESTART (This bit is ORed with the CM4 debug restart bit from MDM-AP control register before connected as input to the core)	core
[6]	MDMAP CTITRIGOUT (MDM-AP status[22]) The ack is driven by MDM AP trigout ack(MDM-AP Control[25].	MDM-AP
[5]	ETMEXTIN[1]	ETM
[4]	ETMEXTIN[0]	ETM
[3]	NVIC IntID[5]	NVIC
[2]	Not used	-
[1]	INTISR[x]	NVIC
[0]	EDBGRQ	Core

20.9.9.2 CA5 CTI Triggers

CA5 CTI Trigger Inputs:

Table 20-12. CA5 Trigger Inputs

Trigger Bit	Source Signal	Source Device
[7]	Not used (tied to 1'b0)	-
[6]	TRIGGER	ETM
[5]	COMMRX = ctiasicctl0_o[5] ? etb_acqcomp : commrx[0]	Core
[4]	COMMTX = ctiasicctl0_o[4] ? etb_full : commtx[0]	Core
[3]	EXTOUT[1]	ETM
[2]	EXTOUT[0]	ETM
[1]	PMUIRQ[n]	Core
[0]	DBGTRIGGER[n]	Core

For CA5 CTI trigin[5:4] the select used for mux control "ctiasicctl[5:4]" is the "ASICCTL[5:4]" provided by the CTI.

CA5 CTI Trigger Outputs:

Table 20-13. CA5 Trigger Outputs

Trigger Bit	Source Signal	Source Device
[7]	DBGRESTRAT (This bit is ORed with the CA5 debug restart bit from MDM-AP control register before connected as input to the core)	Core
[6]	GIC IntID[37]	Core
[5]	Not used	-
[4]	EXTIN[3]	ETM
[3]	EXTIN[2]	ETM
[2]	EXTIN[1]	ETM
[1]	EXTIN[0]	ETM
[0]	EDBGRQ[n]	Core

20.10 Low Power Debug

The device supports two types of low power modes:

1. Module clock gating (STOP mode): If the debug power up request, CxxxPWRUPREQ is high and the system *attempts to enter STOP mode*, the DAP clock and the FCLK continue to run to support core register access and trace. In this case the debug module will have access to core registers but not modules which are clock gated.

   If the device *is in STOP mode*, the debugger can assert the EDBGREQ by writing to the MDM-AP control bits (CM4 Debug Request and CA5 Debug Request). This will wakeup the core and it will move to halted state.

2. Power gated mode (LPSTOP1/2/3): When the device goes into power gated mode the device debug logic is also powered off. The debugger cannot gather any debug data for the duration of low power mode.

   If the device is expected to enter into the power gated mode and the debugger wants to restore debug on recovery, the debugger sets bit[5] of MDM-AP control register (STANDBYDBGREQ).

   The debugger continues to try to reestablish connection to the debug port while the device is in power gated mode. Once the device exits power gated mode, this status is available in MDM-AP status register- bit[10], "Standby mode exit". The debugger can read the status after it regains JTAG access to be sure that power gated mode was exited and not just any other reset.

The debugger is now able to reconfigure the desired debug state. Once done, it sets the Standby mode debug acknowledge (STANDBYDBGACK) – bit[6] in MDM-AP control register to let SRC to de-assert the core reset.

3. Inhibit Sleep: This is another way in which the debugger can handle the system standby modes. The inhibit Sleep is set using the configuration of MDM-AP control register. The debugger can set this bit, check if it is set (if yes the system has not gone into standby while the bit was being set) and then carry on with the debug accesses. Since inhibit sleep bit is set the system cannot move into standby and the debugger can continue with the accesses without risk of losing communication. Once the debugger is done it can reset the bit and allow the standby transitions (in case the system so desire). The same is repeated every time the debugger wants to make debug accesses.

   In case the system goes into standby while the inhibit Sleep was reset and the debugger now wants to make debug accesses, it initiates a write to SET inhibit Sleep and on read back does not get the expected response. Thus it knows that the system is in standby. The debugger can continue this until the device comes out of standby.

20.11 Secured JTAG

The SJC block offers a range of security levels. The SJC blocks the access to debug based on eFuse configuration and authentication. This is summarized in the table below:

S. No.	Name	Fuse used	Comments	Security Level	Flexibility to change the security level	Field Return
1	No Debug	JTAG_SMODE[1:0]	Closed for debug but allows basic JTAG features like IDCODE, BSR …	High	No change possible	The field return is supported using Fuse bit "FIELD_RETURN" This fuse is blown based on an authenticated image. The part which is received as field return is not returned to the customer. The field return part offers complete access to TCU and debug just like the in fab part.
2	Secured JTAG	JTAG_SMODE[1:0]	Authentication based on challenge-Response (C-R)	Medium	Can switch to no debug	

Table continues on the next page...

S. No.	Name	Fuse used	Comments	Security Level	Flexibility to change the security level	Field Return
3	JTAG enabled	JTAG_SMODE[1:0]	Open	No security	Can switch to secured JTAG or No Debug	Full support possible for field return.

20.11.1 Additional Authentication Interface

Authentication interface aims to restrict access to debug and trace functionality. This is done by controlling the ARM debug authentication signals. For more details please refer to CoreSight v1.0 Architecture Specification. These are listed below:

Control Signal name	Description	Controlled by
DBGEN	Invasive Debug Enable	SJC based on security authentication
NIDEN	Non invasive Debug Enable	SJC based on security authentication
SPIDEN	Secure invasive debug enable	control bit In SRC (can be SET by only s/w running in secure world)
SPNIDEN	Secure non-invasive debug enable	control bit In SRC (can be SET by only s/w running in secure world)

While in FAB or for field return all the controls are enabled to allow open access.

The SJC controls the DBGEN and NIDEN based on the security authentication and perceived security incident related to power up. This allows debugger access to non secure coresight components and allows non secure traces to be traced out.

In case debug in the secure world is required the SPIDEN and SPNIDEN needs to be set. This is feasible only by s/w running in the secure world. Refer to id-_Ref306981484 for details.

20.12 Configuration sequence

20.12.1 Halt mode

This section explains how the different debugger types can make accesses to the debug logic.

1. In case of 2 wire CJTAG debugger send the sequence over TMS and TCK to switch CJTAG in 2 wires (refer to JTAG-to-cJTAG change sequence)

   Else

   In case of ARM SWD debugger send the sequence to switch the DAP from JTAG to SWD (refer to JTAG-to-SWD change sequence)

   Else

   Continue with the IEEE 1149.1 4-wire interface.
2. Access ARM DAP Instruction registers.
3. Access "SJC" through JTAG AP.
4. Clear security authentication.
5. Continue with debug access.

20.12.2 Monitor mode

This is an application initiated mode where in debugger is not connected. In this mode the application writes to a bit in the CCM which allows to override CDBGPWRUPREQ and hence un gate the dap clock. In case of monitor mode only the CA5 debug IPs can be configured and used. We cannot use it for CM4 debug.

Chapter 21
Direct Memory Access Controller (eDMA)

21.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The enhanced direct memory access (eDMA) controller is a second-generation module capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source- and destination-address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 32 channels

21.1.1 Block diagram

This diagram illustrates the eDMA module.

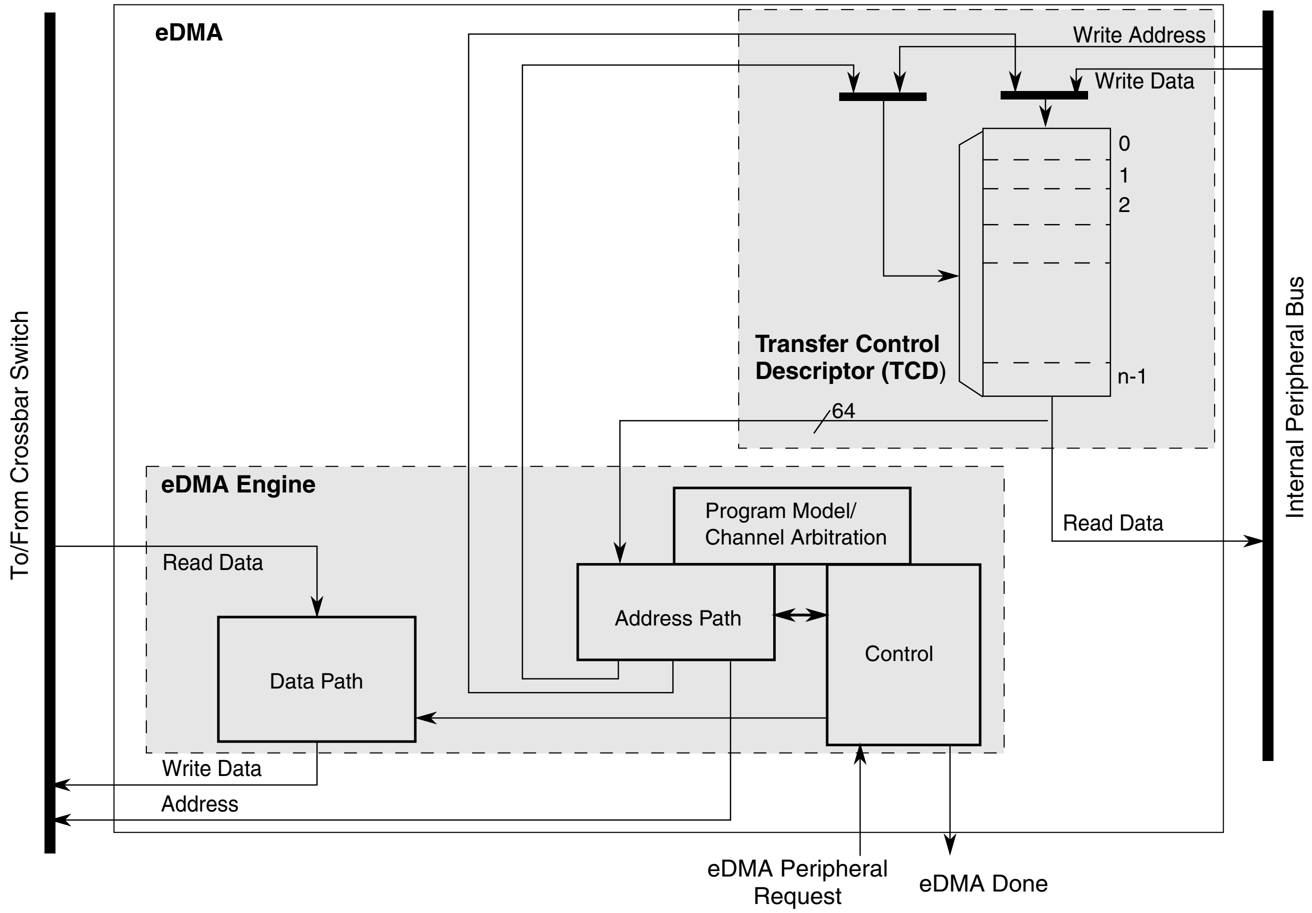


Figure 21-1. eDMA block diagram

21.1.2 Block parts

The eDMA module is partitioned into two major modules: the eDMA engine and the transfer-control descriptor local memory.

The eDMA engine is further partitioned into four submodules:

Table 21-1. eDMA engine submodules

Submodule	Function
Address path	This block implements registered versions of two channel transfer control descriptors, channel x and channel y, and manages all master bus-address calculations. All the channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher priority channel activation is asserted while the first channel is active. After a channel is activated, it runs until the minor loop is completed, unless preempted by a higher priority channel. This provides a mechanism (enabled by DCHPRI*n*[ECP]) where a large data move operation can be preempted to minimize the time another channel is blocked from execution. When any channel is selected to execute, the contents of its TCD are read from local memory and loaded into the address path channel x registers for a normal start and into channel y registers for a preemption start. After the minor loop completes execution, the address path hardware writes the new values for the TCD*n*_{SADDR, DADDR, CITER} back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading the TCD*n*_CITER field, and a possible fetch of the next TCD*n* from memory as part of a scatter/gather operation.
Data path	This block implements the bus master read/write datapath. It includes 16 bytes of register storage and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the 2-stage pipelined internal bus. The address path module represents the 1st stage of the bus pipeline (address phase), while the data path module implements the 2nd stage of the pipeline (data phase).
Program model/channel arbitration	This block implements the first section of the eDMA programming model as well as the channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via control logic).
Control	This block provides all the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/destination write operations until the number of bytes specified in the minor loop byte count has moved. For descriptors where the sizes are not equal, multiple accesses of the smaller size data are required for each reference of the larger size. As an example, if the source size references 16-bit data and the destination is 32-bit data, two reads are performed, then one 32-bit write.

The transfer-control descriptor local memory is further partitioned into:

Table 21-2. Transfer control descriptor memory

Submodule	Description
Memory controller	This logic implements the required dual-ported controller, managing accesses from the eDMA engine as well as references from the internal peripheral bus. As noted earlier, in the event of simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled.
Memory array	TCD storage for each channel's transfer profile.

21.1.3 Features

The eDMA is a highly programmable data-transfer engine optimized to minimize the required intervention from the host processor. It is intended for use in applications where the data size to be transferred is statically known and not defined within the data packet itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination
  - Programmable source and destination addresses and transfer size
  - Support for enhanced addressing modes
- 32-channel implementation that performs complex data transfers with minimal intervention from a host processor
  - Internal data buffer, used as temporary storage to support 16- and 32-byte transfers
  - Connections to the crossbar switch for bus mastering the data movement
- Transfer control descriptor (TCD) organized to support two-deep, nested transfer operations
  - 32-byte TCD stored in local memory for each channel
  - An inner data transfer loop defined by a minor byte transfer count
  - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
  - Explicit software initiation
  - Initiation via a channel-to-channel linking mechanism for continuous transfers
  - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion reported via optional interrupt requests
  - One interrupt per channel, optionally asserted at completion of major iteration count
  - Optional error terminations per channel and logically summed together to form one error interrupt to the interrupt controller
- Optional support for scatter/gather DMA processing

- Support for complex data structures
- Support to cancel transfers via software

In the discussion of this module, *n* is used to reference the channel number.

21.2 Modes of operation

The eDMA operates in the following modes:

Table 21-3. Modes of operation

Mode	Description
Normal	In Normal mode, the eDMA transfers data between a source and a destination. The source and destination can be a memory block or an I/O block capable of operation with the eDMA. A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in the transfer control descriptor (TCD). The minor loop is the sequence of read-write operations that transfers these NBYTES per service request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.
Debug	DMA operation is configurable in Debug mode via the control register: • If CR[EDBG] is cleared, the DMA continues to operate. • If CR[EDBG] is set, the eDMA stops transferring data. If Debug mode is entered while a channel is active, the eDMA continues operation until the channel retires.
Wait	Before entering Wait mode, the DMA attempts to complete its current transfer. After the transfer completes, the device enters Wait mode.

21.3 Memory map/register definition

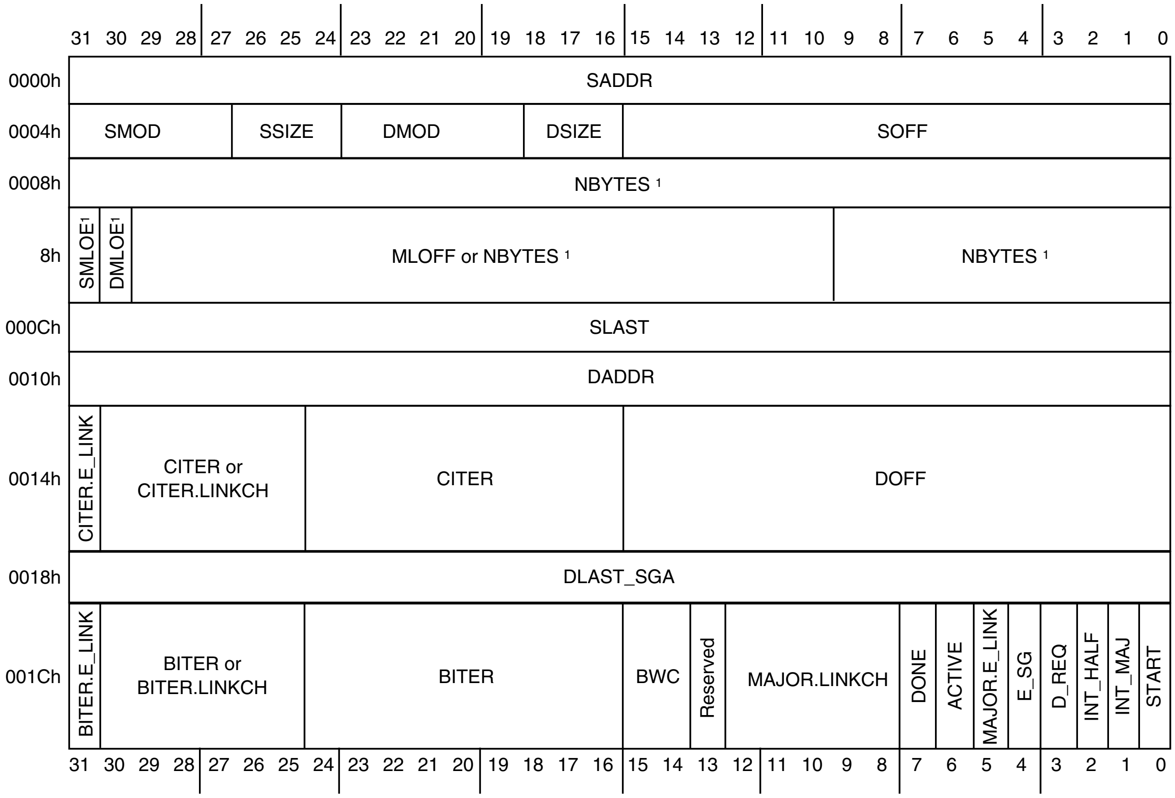
The eDMA's programming model is partitioned into two regions:

- The first region defines a number of registers providing control functions
- The second region corresponds to the local transfer control descriptor (TCD) memory

  Each channel requires a 32-byte transfer control descriptor for defining the desired data movement operation. The channel descriptors are stored in the local memory in sequential order: channel 0, channel 1,... channel 31. Each TCD*n* definition is presented as 11 registers of 16 or 32 bits.

  TCD Initialization: Prior to activating a channel, you must initialize its TCD with the appropriate transfer profile.

Here is the TCD structure:



[1] The fields implemented in Word 2 depend on whether DMA_CR[EMLM] is 0 or 1.

Reserved memory and bit fields:

- Reading reserved bits in a register returns the value of zero.
- Writes to reserved bits in a register are ignored.
- Reading or writing a reserved memory location generates a bus error.

DMA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_8000	Control Register (DMA0_CR)	32	R/W	0000_0400h	21.3.1/990
4001_8004	Error Status Register (DMA0_ES)	32	R	0000_0000h	21.3.2/993
4001_800C	Enable Request Register (DMA0_ERQ)	32	R/W	0000_0000h	21.3.3/995
4001_8014	Enable Error Interrupt Register (DMA0_EEI)	32	R/W	0000_0000h	21.3.4/998
4001_8018	Clear Enable Error Interrupt Register (DMA0_CEEI)	8	W (always reads 0)	00h	21.3.5/1002
4001_8019	Set Enable Error Interrupt Register (DMA0_SEEI)	8	W (always reads 0)	00h	21.3.6/1003
4001_801A	Clear Enable Request Register (DMA0_CERQ)	8	W (always reads 0)	00h	21.3.7/1004
4001_801B	Set Enable Request Register (DMA0_SERQ)	8	W (always reads 0)	00h	21.3.8/1005
4001_801C	Clear DONE Status Bit Register (DMA0_CDNE)	8	W (always reads 0)	00h	21.3.9/1006
4001_801D	Set START Bit Register (DMA0_SSRT)	8	W (always reads 0)	00h	21.3.10/ 1007
4001_801E	Clear Error Register (DMA0_CERR)	8	W (always reads 0)	00h	21.3.11/ 1008
4001_801F	Clear Interrupt Request Register (DMA0_CINT)	8	W (always reads 0)	00h	21.3.12/ 1009
4001_8024	Interrupt Request Register (DMA0_INT)	32	R/W	0000_0000h	21.3.13/ 1009
4001_802C	Error Register (DMA0_ERR)	32	R/W	0000_0000h	21.3.14/ 1013
4001_8034	Hardware Request Status Register (DMA0_HRS)	32	R/W	0000_0000h	21.3.15/ 1017
4001_8044	Enable Asynchronous Request in Stop Register (DMA0_EARS)	32	R/W	0000_0000h	21.3.16/ 1020
4001_8100	Channel n Priority Register (DMA0_DCHPRI3)	8	R/W	See section	21.3.17/ 1024
4001_8101	Channel n Priority Register (DMA0_DCHPRI2)	8	R/W	See section	21.3.17/ 1024

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_8102	Channel n Priority Register (DMA0_DCHPRI1)	8	R/W	See section	21.3.17/ 1024
4001_8103	Channel n Priority Register (DMA0_DCHPRI0)	8	R/W	See section	21.3.17/ 1024
4001_8104	Channel n Priority Register (DMA0_DCHPRI7)	8	R/W	See section	21.3.17/ 1024
4001_8105	Channel n Priority Register (DMA0_DCHPRI6)	8	R/W	See section	21.3.17/ 1024
4001_8106	Channel n Priority Register (DMA0_DCHPRI5)	8	R/W	See section	21.3.17/ 1024
4001_8107	Channel n Priority Register (DMA0_DCHPRI4)	8	R/W	See section	21.3.17/ 1024
4001_8108	Channel n Priority Register (DMA0_DCHPRI11)	8	R/W	See section	21.3.17/ 1024
4001_8109	Channel n Priority Register (DMA0_DCHPRI10)	8	R/W	See section	21.3.17/ 1024
4001_810A	Channel n Priority Register (DMA0_DCHPRI9)	8	R/W	See section	21.3.17/ 1024
4001_810B	Channel n Priority Register (DMA0_DCHPRI8)	8	R/W	See section	21.3.17/ 1024
4001_810C	Channel n Priority Register (DMA0_DCHPRI15)	8	R/W	See section	21.3.17/ 1024
4001_810D	Channel n Priority Register (DMA0_DCHPRI14)	8	R/W	See section	21.3.17/ 1024
4001_810E	Channel n Priority Register (DMA0_DCHPRI13)	8	R/W	See section	21.3.17/ 1024
4001_810F	Channel n Priority Register (DMA0_DCHPRI12)	8	R/W	See section	21.3.17/ 1024
4001_8110	Channel n Priority Register (DMA0_DCHPRI19)	8	R/W	See section	21.3.17/ 1024
4001_8111	Channel n Priority Register (DMA0_DCHPRI18)	8	R/W	See section	21.3.17/ 1024
4001_8112	Channel n Priority Register (DMA0_DCHPRI17)	8	R/W	See section	21.3.17/ 1024
4001_8113	Channel n Priority Register (DMA0_DCHPRI16)	8	R/W	See section	21.3.17/ 1024
4001_8114	Channel n Priority Register (DMA0_DCHPRI23)	8	R/W	See section	21.3.17/ 1024
4001_8115	Channel n Priority Register (DMA0_DCHPRI22)	8	R/W	See section	21.3.17/ 1024
4001_8116	Channel n Priority Register (DMA0_DCHPRI21)	8	R/W	See section	21.3.17/ 1024
4001_8117	Channel n Priority Register (DMA0_DCHPRI20)	8	R/W	See section	21.3.17/ 1024

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_8118	Channel n Priority Register (DMA0_DCHPRI27)	8	R/W	See section	21.3.17/ 1024
4001_8119	Channel n Priority Register (DMA0_DCHPRI26)	8	R/W	See section	21.3.17/ 1024
4001_811A	Channel n Priority Register (DMA0_DCHPRI25)	8	R/W	See section	21.3.17/ 1024
4001_811B	Channel n Priority Register (DMA0_DCHPRI24)	8	R/W	See section	21.3.17/ 1024
4001_811C	Channel n Priority Register (DMA0_DCHPRI31)	8	R/W	See section	21.3.17/ 1024
4001_811D	Channel n Priority Register (DMA0_DCHPRI30)	8	R/W	See section	21.3.17/ 1024
4001_811E	Channel n Priority Register (DMA0_DCHPRI29)	8	R/W	See section	21.3.17/ 1024
4001_811F	Channel n Priority Register (DMA0_DCHPRI28)	8	R/W	See section	21.3.17/ 1024
4001_9000	TCD Source Address (DMA0_TCD0_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9004	TCD Signed Source Address Offset (DMA0_TCD0_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9006	TCD Transfer Attributes (DMA0_TCD0_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9008	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD0_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9008	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD0_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9008	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD0_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_900C	TCD Last Source Address Adjustment (DMA0_TCD0_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9010	TCD Destination Address (DMA0_TCD0_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9014	TCD Signed Destination Address Offset (DMA0_TCD0_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9016	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD0_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9016	DMA0_TCD0_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9018	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD0_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_901C	TCD Control and Status (DMA0_TCD0_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_901E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD0_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_901E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD0_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9020	TCD Source Address (DMA0_TCD1_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9024	TCD Signed Source Address Offset (DMA0_TCD1_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9026	TCD Transfer Attributes (DMA0_TCD1_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9028	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD1_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9028	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD1_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9028	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD1_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_902C	TCD Last Source Address Adjustment (DMA0_TCD1_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9030	TCD Destination Address (DMA0_TCD1_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9034	TCD Signed Destination Address Offset (DMA0_TCD1_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9036	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD1_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9036	DMA0_TCD1_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9038	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD1_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_903C	TCD Control and Status (DMA0_TCD1_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_903E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD1_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_903E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD1_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9040	TCD Source Address (DMA0_TCD2_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9044	TCD Signed Source Address Offset (DMA0_TCD2_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9046	TCD Transfer Attributes (DMA0_TCD2_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9048	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD2_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9048	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD2_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9048	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD2_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_904C	TCD Last Source Address Adjustment (DMA0_TCD2_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9050	TCD Destination Address (DMA0_TCD2_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9054	TCD Signed Destination Address Offset (DMA0_TCD2_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9056	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD2_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9056	DMA0_TCD2_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9058	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD2_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_905C	TCD Control and Status (DMA0_TCD2_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_905E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD2_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_905E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD2_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9060	TCD Source Address (DMA0_TCD3_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9064	TCD Signed Source Address Offset (DMA0_TCD3_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9066	TCD Transfer Attributes (DMA0_TCD3_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9068	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD3_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9068	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD3_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9068	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD3_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_906C	TCD Last Source Address Adjustment (DMA0_TCD3_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9070	TCD Destination Address (DMA0_TCD3_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9074	TCD Signed Destination Address Offset (DMA0_TCD3_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9076	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD3_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9076	DMA0_TCD3_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9078	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD3_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_907C	TCD Control and Status (DMA0_TCD3_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_907E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD3_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_907E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD3_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9080	TCD Source Address (DMA0_TCD4_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9084	TCD Signed Source Address Offset (DMA0_TCD4_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9086	TCD Transfer Attributes (DMA0_TCD4_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9088	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD4_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9088	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD4_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9088	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD4_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_908C	TCD Last Source Address Adjustment (DMA0_TCD4_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9090	TCD Destination Address (DMA0_TCD4_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9094	TCD Signed Destination Address Offset (DMA0_TCD4_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9096	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD4_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9096	DMA0_TCD4_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9098	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD4_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_909C	TCD Control and Status (DMA0_TCD4_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_909E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD4_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_909E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD4_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_90A0	TCD Source Address (DMA0_TCD5_SADDR)	32	R/W	Undefined	21.3.18/ 1025

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_90A4	TCD Signed Source Address Offset (DMA0_TCD5_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_90A6	TCD Transfer Attributes (DMA0_TCD5_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_90A8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD5_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_90A8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD5_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_90A8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD5_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_90AC	TCD Last Source Address Adjustment (DMA0_TCD5_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_90B0	TCD Destination Address (DMA0_TCD5_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_90B4	TCD Signed Destination Address Offset (DMA0_TCD5_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_90B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD5_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_90B6	DMA0_TCD5_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_90B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD5_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_90BC	TCD Control and Status (DMA0_TCD5_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_90BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD5_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_90BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD5_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_90C0	TCD Source Address (DMA0_TCD6_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_90C4	TCD Signed Source Address Offset (DMA0_TCD6_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_90C6	TCD Transfer Attributes (DMA0_TCD6_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_90C8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD6_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_90C8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD6_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_90C8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD6_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_90CC	TCD Last Source Address Adjustment (DMA0_TCD6_SLAST)	32	R/W	Undefined	21.3.24/ 1030

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_90D0	TCD Destination Address (DMA0_TCD6_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_90D4	TCD Signed Destination Address Offset (DMA0_TCD6_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_90D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD6_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_90D6	DMA0_TCD6_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_90D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD6_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_90DC	TCD Control and Status (DMA0_TCD6_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_90DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD6_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_90DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD6_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_90E0	TCD Source Address (DMA0_TCD7_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_90E4	TCD Signed Source Address Offset (DMA0_TCD7_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_90E6	TCD Transfer Attributes (DMA0_TCD7_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_90E8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD7_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_90E8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD7_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_90E8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD7_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_90EC	TCD Last Source Address Adjustment (DMA0_TCD7_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_90F0	TCD Destination Address (DMA0_TCD7_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_90F4	TCD Signed Destination Address Offset (DMA0_TCD7_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_90F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD7_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_90F6	DMA0_TCD7_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_90F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD7_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_90FC	TCD Control and Status (DMA0_TCD7_CSR)	16	R/W	Undefined	21.3.30/ 1034

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_90FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD7_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_90FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD7_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9100	TCD Source Address (DMA0_TCD8_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9104	TCD Signed Source Address Offset (DMA0_TCD8_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9106	TCD Transfer Attributes (DMA0_TCD8_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9108	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD8_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9108	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD8_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9108	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD8_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_910C	TCD Last Source Address Adjustment (DMA0_TCD8_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9110	TCD Destination Address (DMA0_TCD8_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9114	TCD Signed Destination Address Offset (DMA0_TCD8_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9116	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD8_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9116	DMA0_TCD8_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9118	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD8_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_911C	TCD Control and Status (DMA0_TCD8_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_911E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD8_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_911E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD8_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9120	TCD Source Address (DMA0_TCD9_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9124	TCD Signed Source Address Offset (DMA0_TCD9_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9126	TCD Transfer Attributes (DMA0_TCD9_ATTR)	16	R/W	Undefined	21.3.20/ 1026

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9128	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD9_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9128	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD9_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9128	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD9_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_912C	TCD Last Source Address Adjustment (DMA0_TCD9_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9130	TCD Destination Address (DMA0_TCD9_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9134	TCD Signed Destination Address Offset (DMA0_TCD9_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9136	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD9_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9136	DMA0_TCD9_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9138	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD9_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_913C	TCD Control and Status (DMA0_TCD9_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_913E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD9_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_913E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD9_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9140	TCD Source Address (DMA0_TCD10_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9144	TCD Signed Source Address Offset (DMA0_TCD10_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9146	TCD Transfer Attributes (DMA0_TCD10_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9148	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD10_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9148	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD10_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9148	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD10_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_914C	TCD Last Source Address Adjustment (DMA0_TCD10_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9150	TCD Destination Address (DMA0_TCD10_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9154	TCD Signed Destination Address Offset (DMA0_TCD10_DOFF)	16	R/W	Undefined	21.3.26/ 1031

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9156	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD10_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9156	DMA0_TCD10_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9158	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD10_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_915C	TCD Control and Status (DMA0_TCD10_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_915E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD10_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_915E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD10_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9160	TCD Source Address (DMA0_TCD11_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9164	TCD Signed Source Address Offset (DMA0_TCD11_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9166	TCD Transfer Attributes (DMA0_TCD11_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9168	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD11_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9168	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD11_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9168	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD11_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_916C	TCD Last Source Address Adjustment (DMA0_TCD11_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9170	TCD Destination Address (DMA0_TCD11_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9174	TCD Signed Destination Address Offset (DMA0_TCD11_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9176	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD11_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9176	DMA0_TCD11_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9178	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD11_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_917C	TCD Control and Status (DMA0_TCD11_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_917E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD11_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_917E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD11_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9180	TCD Source Address (DMA0_TCD12_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9184	TCD Signed Source Address Offset (DMA0_TCD12_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9186	TCD Transfer Attributes (DMA0_TCD12_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9188	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD12_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9188	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD12_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9188	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD12_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_918C	TCD Last Source Address Adjustment (DMA0_TCD12_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9190	TCD Destination Address (DMA0_TCD12_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9194	TCD Signed Destination Address Offset (DMA0_TCD12_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9196	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD12_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9196	DMA0_TCD12_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9198	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD12_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_919C	TCD Control and Status (DMA0_TCD12_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_919E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD12_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_919E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD12_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_91A0	TCD Source Address (DMA0_TCD13_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_91A4	TCD Signed Source Address Offset (DMA0_TCD13_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_91A6	TCD Transfer Attributes (DMA0_TCD13_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_91A8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD13_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_91A8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD13_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_91A8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD13_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_91AC	TCD Last Source Address Adjustment (DMA0_TCD13_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_91B0	TCD Destination Address (DMA0_TCD13_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_91B4	TCD Signed Destination Address Offset (DMA0_TCD13_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_91B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD13_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_91B6	DMA0_TCD13_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_91B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD13_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_91BC	TCD Control and Status (DMA0_TCD13_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_91BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD13_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_91BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD13_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_91C0	TCD Source Address (DMA0_TCD14_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_91C4	TCD Signed Source Address Offset (DMA0_TCD14_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_91C6	TCD Transfer Attributes (DMA0_TCD14_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_91C8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD14_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_91C8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD14_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_91C8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD14_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_91CC	TCD Last Source Address Adjustment (DMA0_TCD14_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_91D0	TCD Destination Address (DMA0_TCD14_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_91D4	TCD Signed Destination Address Offset (DMA0_TCD14_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_91D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD14_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_91D6	DMA0_TCD14_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_91D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD14_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_91DC	TCD Control and Status (DMA0_TCD14_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_91DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD14_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_91DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD14_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_91E0	TCD Source Address (DMA0_TCD15_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_91E4	TCD Signed Source Address Offset (DMA0_TCD15_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_91E6	TCD Transfer Attributes (DMA0_TCD15_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_91E8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD15_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_91E8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD15_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_91E8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD15_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_91EC	TCD Last Source Address Adjustment (DMA0_TCD15_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_91F0	TCD Destination Address (DMA0_TCD15_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_91F4	TCD Signed Destination Address Offset (DMA0_TCD15_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_91F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD15_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_91F6	DMA0_TCD15_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_91F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD15_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_91FC	TCD Control and Status (DMA0_TCD15_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_91FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD15_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_91FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD15_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9200	TCD Source Address (DMA0_TCD16_SADDR)	32	R/W	Undefined	21.3.18/ 1025

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9204	TCD Signed Source Address Offset (DMA0_TCD16_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9206	TCD Transfer Attributes (DMA0_TCD16_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9208	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD16_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9208	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD16_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9208	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD16_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_920C	TCD Last Source Address Adjustment (DMA0_TCD16_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9210	TCD Destination Address (DMA0_TCD16_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9214	TCD Signed Destination Address Offset (DMA0_TCD16_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9216	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD16_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9216	DMA0_TCD16_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9218	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD16_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_921C	TCD Control and Status (DMA0_TCD16_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_921E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD16_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_921E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD16_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9220	TCD Source Address (DMA0_TCD17_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9224	TCD Signed Source Address Offset (DMA0_TCD17_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9226	TCD Transfer Attributes (DMA0_TCD17_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9228	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD17_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9228	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD17_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9228	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD17_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_922C	TCD Last Source Address Adjustment (DMA0_TCD17_SLAST)	32	R/W	Undefined	21.3.24/ 1030

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9230	TCD Destination Address (DMA0_TCD17_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9234	TCD Signed Destination Address Offset (DMA0_TCD17_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9236	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD17_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9236	DMA0_TCD17_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9238	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD17_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_923C	TCD Control and Status (DMA0_TCD17_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_923E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD17_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_923E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD17_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9240	TCD Source Address (DMA0_TCD18_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9244	TCD Signed Source Address Offset (DMA0_TCD18_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9246	TCD Transfer Attributes (DMA0_TCD18_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9248	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD18_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9248	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD18_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9248	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD18_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_924C	TCD Last Source Address Adjustment (DMA0_TCD18_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9250	TCD Destination Address (DMA0_TCD18_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9254	TCD Signed Destination Address Offset (DMA0_TCD18_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9256	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD18_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9256	DMA0_TCD18_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9258	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD18_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_925C	TCD Control and Status (DMA0_TCD18_CSR)	16	R/W	Undefined	21.3.30/ 1034

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_925E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD18_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_925E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD18_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9260	TCD Source Address (DMA0_TCD19_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9264	TCD Signed Source Address Offset (DMA0_TCD19_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9266	TCD Transfer Attributes (DMA0_TCD19_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9268	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD19_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9268	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD19_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9268	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD19_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_926C	TCD Last Source Address Adjustment (DMA0_TCD19_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9270	TCD Destination Address (DMA0_TCD19_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9274	TCD Signed Destination Address Offset (DMA0_TCD19_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9276	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD19_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9276	DMA0_TCD19_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9278	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD19_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_927C	TCD Control and Status (DMA0_TCD19_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_927E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD19_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_927E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD19_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9280	TCD Source Address (DMA0_TCD20_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9284	TCD Signed Source Address Offset (DMA0_TCD20_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9286	TCD Transfer Attributes (DMA0_TCD20_ATTR)	16	R/W	Undefined	21.3.20/ 1026

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9288	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD20_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9288	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD20_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9288	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD20_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_928C	TCD Last Source Address Adjustment (DMA0_TCD20_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9290	TCD Destination Address (DMA0_TCD20_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9294	TCD Signed Destination Address Offset (DMA0_TCD20_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9296	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD20_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9296	DMA0_TCD20_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9298	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD20_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_929C	TCD Control and Status (DMA0_TCD20_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_929E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD20_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_929E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD20_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_92A0	TCD Source Address (DMA0_TCD21_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_92A4	TCD Signed Source Address Offset (DMA0_TCD21_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_92A6	TCD Transfer Attributes (DMA0_TCD21_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_92A8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD21_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_92A8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD21_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_92A8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD21_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_92AC	TCD Last Source Address Adjustment (DMA0_TCD21_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_92B0	TCD Destination Address (DMA0_TCD21_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_92B4	TCD Signed Destination Address Offset (DMA0_TCD21_DOFF)	16	R/W	Undefined	21.3.26/ 1031

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_92B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD21_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_92B6	DMA0_TCD21_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_92B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD21_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_92BC	TCD Control and Status (DMA0_TCD21_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_92BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD21_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_92BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD21_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_92C0	TCD Source Address (DMA0_TCD22_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_92C4	TCD Signed Source Address Offset (DMA0_TCD22_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_92C6	TCD Transfer Attributes (DMA0_TCD22_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_92C8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD22_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_92C8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD22_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_92C8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD22_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_92CC	TCD Last Source Address Adjustment (DMA0_TCD22_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_92D0	TCD Destination Address (DMA0_TCD22_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_92D4	TCD Signed Destination Address Offset (DMA0_TCD22_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_92D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD22_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_92D6	DMA0_TCD22_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_92D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD22_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_92DC	TCD Control and Status (DMA0_TCD22_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_92DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD22_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_92DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD22_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_92E0	TCD Source Address (DMA0_TCD23_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_92E4	TCD Signed Source Address Offset (DMA0_TCD23_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_92E6	TCD Transfer Attributes (DMA0_TCD23_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_92E8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD23_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_92E8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD23_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_92E8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD23_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_92EC	TCD Last Source Address Adjustment (DMA0_TCD23_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_92F0	TCD Destination Address (DMA0_TCD23_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_92F4	TCD Signed Destination Address Offset (DMA0_TCD23_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_92F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD23_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_92F6	DMA0_TCD23_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_92F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD23_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_92FC	TCD Control and Status (DMA0_TCD23_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_92FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD23_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_92FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD23_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9300	TCD Source Address (DMA0_TCD24_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9304	TCD Signed Source Address Offset (DMA0_TCD24_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9306	TCD Transfer Attributes (DMA0_TCD24_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9308	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD24_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9308	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD24_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9308	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD24_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_930C	TCD Last Source Address Adjustment (DMA0_TCD24_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9310	TCD Destination Address (DMA0_TCD24_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9314	TCD Signed Destination Address Offset (DMA0_TCD24_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9316	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD24_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9316	DMA0_TCD24_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9318	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD24_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_931C	TCD Control and Status (DMA0_TCD24_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_931E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD24_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_931E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD24_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9320	TCD Source Address (DMA0_TCD25_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9324	TCD Signed Source Address Offset (DMA0_TCD25_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9326	TCD Transfer Attributes (DMA0_TCD25_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9328	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD25_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9328	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD25_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9328	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD25_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_932C	TCD Last Source Address Adjustment (DMA0_TCD25_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9330	TCD Destination Address (DMA0_TCD25_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9334	TCD Signed Destination Address Offset (DMA0_TCD25_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9336	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD25_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9336	DMA0_TCD25_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9338	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD25_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_933C	TCD Control and Status (DMA0_TCD25_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_933E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD25_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_933E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD25_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9340	TCD Source Address (DMA0_TCD26_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9344	TCD Signed Source Address Offset (DMA0_TCD26_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9346	TCD Transfer Attributes (DMA0_TCD26_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9348	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD26_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9348	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD26_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9348	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD26_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_934C	TCD Last Source Address Adjustment (DMA0_TCD26_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9350	TCD Destination Address (DMA0_TCD26_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9354	TCD Signed Destination Address Offset (DMA0_TCD26_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9356	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD26_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9356	DMA0_TCD26_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9358	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD26_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_935C	TCD Control and Status (DMA0_TCD26_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_935E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD26_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_935E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD26_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9360	TCD Source Address (DMA0_TCD27_SADDR)	32	R/W	Undefined	21.3.18/ 1025

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9364	TCD Signed Source Address Offset (DMA0_TCD27_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9366	TCD Transfer Attributes (DMA0_TCD27_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9368	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD27_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9368	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD27_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9368	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD27_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_936C	TCD Last Source Address Adjustment (DMA0_TCD27_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_9370	TCD Destination Address (DMA0_TCD27_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9374	TCD Signed Destination Address Offset (DMA0_TCD27_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9376	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD27_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9376	DMA0_TCD27_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9378	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD27_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_937C	TCD Control and Status (DMA0_TCD27_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_937E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD27_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_937E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD27_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_9380	TCD Source Address (DMA0_TCD28_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_9384	TCD Signed Source Address Offset (DMA0_TCD28_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_9386	TCD Transfer Attributes (DMA0_TCD28_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_9388	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD28_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_9388	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD28_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_9388	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD28_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_938C	TCD Last Source Address Adjustment (DMA0_TCD28_SLAST)	32	R/W	Undefined	21.3.24/ 1030

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_9390	TCD Destination Address (DMA0_TCD28_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_9394	TCD Signed Destination Address Offset (DMA0_TCD28_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_9396	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD28_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_9396	DMA0_TCD28_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_9398	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD28_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_939C	TCD Control and Status (DMA0_TCD28_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_939E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD28_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_939E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD28_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_93A0	TCD Source Address (DMA0_TCD29_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_93A4	TCD Signed Source Address Offset (DMA0_TCD29_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_93A6	TCD Transfer Attributes (DMA0_TCD29_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_93A8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD29_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_93A8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD29_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_93A8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD29_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_93AC	TCD Last Source Address Adjustment (DMA0_TCD29_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_93B0	TCD Destination Address (DMA0_TCD29_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_93B4	TCD Signed Destination Address Offset (DMA0_TCD29_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_93B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD29_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_93B6	DMA0_TCD29_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_93B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD29_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_93BC	TCD Control and Status (DMA0_TCD29_CSR)	16	R/W	Undefined	21.3.30/ 1034

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_93BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD29_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_93BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD29_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_93C0	TCD Source Address (DMA0_TCD30_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_93C4	TCD Signed Source Address Offset (DMA0_TCD30_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_93C6	TCD Transfer Attributes (DMA0_TCD30_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4001_93C8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD30_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_93C8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD30_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_93C8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD30_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_93CC	TCD Last Source Address Adjustment (DMA0_TCD30_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_93D0	TCD Destination Address (DMA0_TCD30_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_93D4	TCD Signed Destination Address Offset (DMA0_TCD30_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_93D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD30_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_93D6	DMA0_TCD30_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_93D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD30_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_93DC	TCD Control and Status (DMA0_TCD30_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_93DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD30_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_93DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD30_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4001_93E0	TCD Source Address (DMA0_TCD31_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4001_93E4	TCD Signed Source Address Offset (DMA0_TCD31_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4001_93E6	TCD Transfer Attributes (DMA0_TCD31_ATTR)	16	R/W	Undefined	21.3.20/ 1026

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_93E8	TCD Minor Byte Count (Minor Loop Disabled) (DMA0_TCD31_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4001_93E8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA0_TCD31_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4001_93E8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA0_TCD31_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4001_93EC	TCD Last Source Address Adjustment (DMA0_TCD31_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4001_93F0	TCD Destination Address (DMA0_TCD31_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4001_93F4	TCD Signed Destination Address Offset (DMA0_TCD31_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4001_93F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD31_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4001_93F6	DMA0_TCD31_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4001_93F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA0_TCD31_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4001_93FC	TCD Control and Status (DMA0_TCD31_CSR)	16	R/W	Undefined	21.3.30/ 1034
4001_93FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA0_TCD31_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4001_93FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA0_TCD31_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_8000	Control Register (DMA1_CR)	32	R/W	0000_0400h	21.3.1/990
4009_8004	Error Status Register (DMA1_ES)	32	R	0000_0000h	21.3.2/993
4009_800C	Enable Request Register (DMA1_ERQ)	32	R/W	0000_0000h	21.3.3/995
4009_8014	Enable Error Interrupt Register (DMA1_EEI)	32	R/W	0000_0000h	21.3.4/998
4009_8018	Clear Enable Error Interrupt Register (DMA1_CEEI)	8	W (always reads 0)	00h	21.3.5/1002
4009_8019	Set Enable Error Interrupt Register (DMA1_SEEI)	8	W (always reads 0)	00h	21.3.6/1003
4009_801A	Clear Enable Request Register (DMA1_CERQ)	8	W (always reads 0)	00h	21.3.7/1004
4009_801B	Set Enable Request Register (DMA1_SERQ)	8	W (always reads 0)	00h	21.3.8/1005
4009_801C	Clear DONE Status Bit Register (DMA1_CDNE)	8	W (always reads 0)	00h	21.3.9/1006

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_801D	Set START Bit Register (DMA1_SSRT)	8	W (always reads 0)	00h	21.3.10/ 1007
4009_801E	Clear Error Register (DMA1_CERR)	8	W (always reads 0)	00h	21.3.11/ 1008
4009_801F	Clear Interrupt Request Register (DMA1_CINT)	8	W (always reads 0)	00h	21.3.12/ 1009
4009_8024	Interrupt Request Register (DMA1_INT)	32	R/W	0000_0000h	21.3.13/ 1009
4009_802C	Error Register (DMA1_ERR)	32	R/W	0000_0000h	21.3.14/ 1013
4009_8034	Hardware Request Status Register (DMA1_HRS)	32	R/W	0000_0000h	21.3.15/ 1017
4009_8044	Enable Asynchronous Request in Stop Register (DMA1_EARS)	32	R/W	0000_0000h	21.3.16/ 1020
4009_8100	Channel n Priority Register (DMA1_DCHPRI3)	8	R/W	See section	21.3.17/ 1024
4009_8101	Channel n Priority Register (DMA1_DCHPRI2)	8	R/W	See section	21.3.17/ 1024
4009_8102	Channel n Priority Register (DMA1_DCHPRI1)	8	R/W	See section	21.3.17/ 1024
4009_8103	Channel n Priority Register (DMA1_DCHPRI0)	8	R/W	See section	21.3.17/ 1024
4009_8104	Channel n Priority Register (DMA1_DCHPRI7)	8	R/W	See section	21.3.17/ 1024
4009_8105	Channel n Priority Register (DMA1_DCHPRI6)	8	R/W	See section	21.3.17/ 1024
4009_8106	Channel n Priority Register (DMA1_DCHPRI5)	8	R/W	See section	21.3.17/ 1024
4009_8107	Channel n Priority Register (DMA1_DCHPRI4)	8	R/W	See section	21.3.17/ 1024
4009_8108	Channel n Priority Register (DMA1_DCHPRI11)	8	R/W	See section	21.3.17/ 1024
4009_8109	Channel n Priority Register (DMA1_DCHPRI10)	8	R/W	See section	21.3.17/ 1024
4009_810A	Channel n Priority Register (DMA1_DCHPRI9)	8	R/W	See section	21.3.17/ 1024
4009_810B	Channel n Priority Register (DMA1_DCHPRI8)	8	R/W	See section	21.3.17/ 1024
4009_810C	Channel n Priority Register (DMA1_DCHPRI15)	8	R/W	See section	21.3.17/ 1024
4009_810D	Channel n Priority Register (DMA1_DCHPRI14)	8	R/W	See section	21.3.17/ 1024

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_810E	Channel n Priority Register (DMA1_DCHPRI13)	8	R/W	See section	21.3.17/ 1024
4009_810F	Channel n Priority Register (DMA1_DCHPRI12)	8	R/W	See section	21.3.17/ 1024
4009_8110	Channel n Priority Register (DMA1_DCHPRI19)	8	R/W	See section	21.3.17/ 1024
4009_8111	Channel n Priority Register (DMA1_DCHPRI18)	8	R/W	See section	21.3.17/ 1024
4009_8112	Channel n Priority Register (DMA1_DCHPRI17)	8	R/W	See section	21.3.17/ 1024
4009_8113	Channel n Priority Register (DMA1_DCHPRI16)	8	R/W	See section	21.3.17/ 1024
4009_8114	Channel n Priority Register (DMA1_DCHPRI23)	8	R/W	See section	21.3.17/ 1024
4009_8115	Channel n Priority Register (DMA1_DCHPRI22)	8	R/W	See section	21.3.17/ 1024
4009_8116	Channel n Priority Register (DMA1_DCHPRI21)	8	R/W	See section	21.3.17/ 1024
4009_8117	Channel n Priority Register (DMA1_DCHPRI20)	8	R/W	See section	21.3.17/ 1024
4009_8118	Channel n Priority Register (DMA1_DCHPRI27)	8	R/W	See section	21.3.17/ 1024
4009_8119	Channel n Priority Register (DMA1_DCHPRI26)	8	R/W	See section	21.3.17/ 1024
4009_811A	Channel n Priority Register (DMA1_DCHPRI25)	8	R/W	See section	21.3.17/ 1024
4009_811B	Channel n Priority Register (DMA1_DCHPRI24)	8	R/W	See section	21.3.17/ 1024
4009_811C	Channel n Priority Register (DMA1_DCHPRI31)	8	R/W	See section	21.3.17/ 1024
4009_811D	Channel n Priority Register (DMA1_DCHPRI30)	8	R/W	See section	21.3.17/ 1024
4009_811E	Channel n Priority Register (DMA1_DCHPRI29)	8	R/W	See section	21.3.17/ 1024
4009_811F	Channel n Priority Register (DMA1_DCHPRI28)	8	R/W	See section	21.3.17/ 1024
4009_9000	TCD Source Address (DMA1_TCD0_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9004	TCD Signed Source Address Offset (DMA1_TCD0_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9006	TCD Transfer Attributes (DMA1_TCD0_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9008	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD0_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9008	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD0_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9008	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD0_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_900C	TCD Last Source Address Adjustment (DMA1_TCD0_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9010	TCD Destination Address (DMA1_TCD0_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9014	TCD Signed Destination Address Offset (DMA1_TCD0_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9016	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD0_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9016	DMA1_TCD0_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9018	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD0_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_901C	TCD Control and Status (DMA1_TCD0_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_901E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD0_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_901E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD0_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9020	TCD Source Address (DMA1_TCD1_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9024	TCD Signed Source Address Offset (DMA1_TCD1_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9026	TCD Transfer Attributes (DMA1_TCD1_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9028	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD1_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9028	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD1_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9028	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD1_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_902C	TCD Last Source Address Adjustment (DMA1_TCD1_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9030	TCD Destination Address (DMA1_TCD1_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9034	TCD Signed Destination Address Offset (DMA1_TCD1_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9036	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD1_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9036	DMA1_TCD1_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9038	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD1_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_903C	TCD Control and Status (DMA1_TCD1_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_903E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD1_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_903E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD1_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9040	TCD Source Address (DMA1_TCD2_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9044	TCD Signed Source Address Offset (DMA1_TCD2_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9046	TCD Transfer Attributes (DMA1_TCD2_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9048	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD2_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9048	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD2_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9048	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD2_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_904C	TCD Last Source Address Adjustment (DMA1_TCD2_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9050	TCD Destination Address (DMA1_TCD2_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9054	TCD Signed Destination Address Offset (DMA1_TCD2_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9056	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD2_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9056	DMA1_TCD2_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9058	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD2_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_905C	TCD Control and Status (DMA1_TCD2_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_905E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD2_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_905E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD2_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9060	TCD Source Address (DMA1_TCD3_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9064	TCD Signed Source Address Offset (DMA1_TCD3_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9066	TCD Transfer Attributes (DMA1_TCD3_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9068	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD3_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9068	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD3_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9068	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD3_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_906C	TCD Last Source Address Adjustment (DMA1_TCD3_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9070	TCD Destination Address (DMA1_TCD3_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9074	TCD Signed Destination Address Offset (DMA1_TCD3_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9076	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD3_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9076	DMA1_TCD3_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9078	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD3_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_907C	TCD Control and Status (DMA1_TCD3_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_907E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD3_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_907E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD3_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9080	TCD Source Address (DMA1_TCD4_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9084	TCD Signed Source Address Offset (DMA1_TCD4_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9086	TCD Transfer Attributes (DMA1_TCD4_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9088	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD4_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9088	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD4_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9088	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD4_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_908C	TCD Last Source Address Adjustment (DMA1_TCD4_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9090	TCD Destination Address (DMA1_TCD4_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9094	TCD Signed Destination Address Offset (DMA1_TCD4_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9096	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD4_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9096	DMA1_TCD4_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9098	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD4_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_909C	TCD Control and Status (DMA1_TCD4_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_909E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD4_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_909E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD4_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_90A0	TCD Source Address (DMA1_TCD5_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_90A4	TCD Signed Source Address Offset (DMA1_TCD5_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_90A6	TCD Transfer Attributes (DMA1_TCD5_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_90A8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD5_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_90A8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD5_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_90A8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD5_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_90AC	TCD Last Source Address Adjustment (DMA1_TCD5_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_90B0	TCD Destination Address (DMA1_TCD5_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_90B4	TCD Signed Destination Address Offset (DMA1_TCD5_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_90B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD5_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_90B6	DMA1_TCD5_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_90B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD5_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_90BC	TCD Control and Status (DMA1_TCD5_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_90BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD5_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_90BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD5_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_90C0	TCD Source Address (DMA1_TCD6_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_90C4	TCD Signed Source Address Offset (DMA1_TCD6_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_90C6	TCD Transfer Attributes (DMA1_TCD6_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_90C8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD6_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_90C8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD6_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_90C8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD6_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_90CC	TCD Last Source Address Adjustment (DMA1_TCD6_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_90D0	TCD Destination Address (DMA1_TCD6_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_90D4	TCD Signed Destination Address Offset (DMA1_TCD6_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_90D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD6_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_90D6	DMA1_TCD6_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_90D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD6_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_90DC	TCD Control and Status (DMA1_TCD6_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_90DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD6_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_90DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD6_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_90E0	TCD Source Address (DMA1_TCD7_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_90E4	TCD Signed Source Address Offset (DMA1_TCD7_SOFF)	16	R/W	Undefined	21.3.19/ 1025

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_90E6	TCD Transfer Attributes (DMA1_TCD7_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_90E8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD7_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_90E8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD7_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_90E8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD7_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_90EC	TCD Last Source Address Adjustment (DMA1_TCD7_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_90F0	TCD Destination Address (DMA1_TCD7_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_90F4	TCD Signed Destination Address Offset (DMA1_TCD7_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_90F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD7_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_90F6	DMA1_TCD7_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_90F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD7_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_90FC	TCD Control and Status (DMA1_TCD7_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_90FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD7_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_90FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD7_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9100	TCD Source Address (DMA1_TCD8_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9104	TCD Signed Source Address Offset (DMA1_TCD8_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9106	TCD Transfer Attributes (DMA1_TCD8_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9108	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD8_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9108	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD8_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9108	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD8_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_910C	TCD Last Source Address Adjustment (DMA1_TCD8_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9110	TCD Destination Address (DMA1_TCD8_DADDR)	32	R/W	Undefined	21.3.25/ 1030

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9114	TCD Signed Destination Address Offset (DMA1_TCD8_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9116	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD8_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9116	DMA1_TCD8_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9118	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD8_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_911C	TCD Control and Status (DMA1_TCD8_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_911E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD8_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_911E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD8_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9120	TCD Source Address (DMA1_TCD9_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9124	TCD Signed Source Address Offset (DMA1_TCD9_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9126	TCD Transfer Attributes (DMA1_TCD9_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9128	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD9_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9128	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD9_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9128	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD9_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_912C	TCD Last Source Address Adjustment (DMA1_TCD9_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9130	TCD Destination Address (DMA1_TCD9_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9134	TCD Signed Destination Address Offset (DMA1_TCD9_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9136	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD9_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9136	DMA1_TCD9_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9138	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD9_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_913C	TCD Control and Status (DMA1_TCD9_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_913E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD9_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_913E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD9_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9140	TCD Source Address (DMA1_TCD10_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9144	TCD Signed Source Address Offset (DMA1_TCD10_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9146	TCD Transfer Attributes (DMA1_TCD10_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9148	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD10_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9148	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD10_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9148	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD10_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_914C	TCD Last Source Address Adjustment (DMA1_TCD10_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9150	TCD Destination Address (DMA1_TCD10_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9154	TCD Signed Destination Address Offset (DMA1_TCD10_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9156	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD10_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9156	DMA1_TCD10_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9158	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD10_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_915C	TCD Control and Status (DMA1_TCD10_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_915E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD10_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_915E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD10_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9160	TCD Source Address (DMA1_TCD11_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9164	TCD Signed Source Address Offset (DMA1_TCD11_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9166	TCD Transfer Attributes (DMA1_TCD11_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9168	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD11_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9168	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD11_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9168	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD11_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_916C	TCD Last Source Address Adjustment (DMA1_TCD11_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9170	TCD Destination Address (DMA1_TCD11_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9174	TCD Signed Destination Address Offset (DMA1_TCD11_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9176	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD11_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9176	DMA1_TCD11_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9178	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD11_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_917C	TCD Control and Status (DMA1_TCD11_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_917E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD11_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_917E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD11_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9180	TCD Source Address (DMA1_TCD12_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9184	TCD Signed Source Address Offset (DMA1_TCD12_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9186	TCD Transfer Attributes (DMA1_TCD12_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9188	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD12_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9188	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD12_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9188	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD12_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_918C	TCD Last Source Address Adjustment (DMA1_TCD12_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9190	TCD Destination Address (DMA1_TCD12_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9194	TCD Signed Destination Address Offset (DMA1_TCD12_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9196	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD12_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9196	DMA1_TCD12_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9198	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD12_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_919C	TCD Control and Status (DMA1_TCD12_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_919E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD12_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_919E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD12_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_91A0	TCD Source Address (DMA1_TCD13_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_91A4	TCD Signed Source Address Offset (DMA1_TCD13_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_91A6	TCD Transfer Attributes (DMA1_TCD13_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_91A8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD13_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_91A8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD13_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_91A8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD13_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_91AC	TCD Last Source Address Adjustment (DMA1_TCD13_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_91B0	TCD Destination Address (DMA1_TCD13_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_91B4	TCD Signed Destination Address Offset (DMA1_TCD13_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_91B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD13_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_91B6	DMA1_TCD13_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_91B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD13_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_91BC	TCD Control and Status (DMA1_TCD13_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_91BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD13_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_91BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD13_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_91C0	TCD Source Address (DMA1_TCD14_SADDR)	32	R/W	Undefined	21.3.18/ 1025

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_91C4	TCD Signed Source Address Offset (DMA1_TCD14_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_91C6	TCD Transfer Attributes (DMA1_TCD14_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_91C8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD14_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_91C8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD14_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_91C8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD14_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_91CC	TCD Last Source Address Adjustment (DMA1_TCD14_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_91D0	TCD Destination Address (DMA1_TCD14_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_91D4	TCD Signed Destination Address Offset (DMA1_TCD14_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_91D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD14_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_91D6	DMA1_TCD14_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_91D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD14_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_91DC	TCD Control and Status (DMA1_TCD14_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_91DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD14_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_91DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD14_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_91E0	TCD Source Address (DMA1_TCD15_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_91E4	TCD Signed Source Address Offset (DMA1_TCD15_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_91E6	TCD Transfer Attributes (DMA1_TCD15_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_91E8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD15_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_91E8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD15_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_91E8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD15_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_91EC	TCD Last Source Address Adjustment (DMA1_TCD15_SLAST)	32	R/W	Undefined	21.3.24/ 1030

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_91F0	TCD Destination Address (DMA1_TCD15_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_91F4	TCD Signed Destination Address Offset (DMA1_TCD15_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_91F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD15_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_91F6	DMA1_TCD15_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_91F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD15_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_91FC	TCD Control and Status (DMA1_TCD15_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_91FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD15_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_91FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD15_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9200	TCD Source Address (DMA1_TCD16_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9204	TCD Signed Source Address Offset (DMA1_TCD16_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9206	TCD Transfer Attributes (DMA1_TCD16_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9208	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD16_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9208	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD16_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9208	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD16_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_920C	TCD Last Source Address Adjustment (DMA1_TCD16_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9210	TCD Destination Address (DMA1_TCD16_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9214	TCD Signed Destination Address Offset (DMA1_TCD16_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9216	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD16_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9216	DMA1_TCD16_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9218	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD16_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_921C	TCD Control and Status (DMA1_TCD16_CSR)	16	R/W	Undefined	21.3.30/ 1034

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_921E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD16_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_921E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD16_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9220	TCD Source Address (DMA1_TCD17_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9224	TCD Signed Source Address Offset (DMA1_TCD17_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9226	TCD Transfer Attributes (DMA1_TCD17_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9228	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD17_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9228	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD17_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9228	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD17_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_922C	TCD Last Source Address Adjustment (DMA1_TCD17_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9230	TCD Destination Address (DMA1_TCD17_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9234	TCD Signed Destination Address Offset (DMA1_TCD17_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9236	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD17_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9236	DMA1_TCD17_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9238	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD17_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_923C	TCD Control and Status (DMA1_TCD17_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_923E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD17_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_923E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD17_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9240	TCD Source Address (DMA1_TCD18_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9244	TCD Signed Source Address Offset (DMA1_TCD18_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9246	TCD Transfer Attributes (DMA1_TCD18_ATTR)	16	R/W	Undefined	21.3.20/ 1026

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9248	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD18_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9248	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD18_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9248	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD18_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_924C	TCD Last Source Address Adjustment (DMA1_TCD18_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9250	TCD Destination Address (DMA1_TCD18_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9254	TCD Signed Destination Address Offset (DMA1_TCD18_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9256	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD18_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9256	DMA1_TCD18_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9258	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD18_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_925C	TCD Control and Status (DMA1_TCD18_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_925E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD18_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_925E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD18_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9260	TCD Source Address (DMA1_TCD19_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9264	TCD Signed Source Address Offset (DMA1_TCD19_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9266	TCD Transfer Attributes (DMA1_TCD19_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9268	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD19_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9268	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD19_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9268	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD19_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_926C	TCD Last Source Address Adjustment (DMA1_TCD19_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9270	TCD Destination Address (DMA1_TCD19_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9274	TCD Signed Destination Address Offset (DMA1_TCD19_DOFF)	16	R/W	Undefined	21.3.26/ 1031

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9276	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD19_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9276	DMA1_TCD19_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9278	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD19_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_927C	TCD Control and Status (DMA1_TCD19_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_927E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD19_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_927E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD19_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9280	TCD Source Address (DMA1_TCD20_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9284	TCD Signed Source Address Offset (DMA1_TCD20_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9286	TCD Transfer Attributes (DMA1_TCD20_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9288	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD20_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9288	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD20_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9288	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD20_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_928C	TCD Last Source Address Adjustment (DMA1_TCD20_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9290	TCD Destination Address (DMA1_TCD20_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9294	TCD Signed Destination Address Offset (DMA1_TCD20_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9296	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD20_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9296	DMA1_TCD20_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9298	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD20_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_929C	TCD Control and Status (DMA1_TCD20_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_929E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD20_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_929E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD20_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_92A0	TCD Source Address (DMA1_TCD21_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_92A4	TCD Signed Source Address Offset (DMA1_TCD21_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_92A6	TCD Transfer Attributes (DMA1_TCD21_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_92A8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD21_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_92A8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD21_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_92A8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD21_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_92AC	TCD Last Source Address Adjustment (DMA1_TCD21_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_92B0	TCD Destination Address (DMA1_TCD21_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_92B4	TCD Signed Destination Address Offset (DMA1_TCD21_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_92B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD21_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_92B6	DMA1_TCD21_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_92B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD21_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_92BC	TCD Control and Status (DMA1_TCD21_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_92BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD21_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_92BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD21_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_92C0	TCD Source Address (DMA1_TCD22_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_92C4	TCD Signed Source Address Offset (DMA1_TCD22_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_92C6	TCD Transfer Attributes (DMA1_TCD22_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_92C8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD22_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_92C8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD22_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_92C8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD22_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_92CC	TCD Last Source Address Adjustment (DMA1_TCD22_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_92D0	TCD Destination Address (DMA1_TCD22_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_92D4	TCD Signed Destination Address Offset (DMA1_TCD22_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_92D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD22_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_92D6	DMA1_TCD22_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_92D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD22_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_92DC	TCD Control and Status (DMA1_TCD22_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_92DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD22_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_92DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD22_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_92E0	TCD Source Address (DMA1_TCD23_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_92E4	TCD Signed Source Address Offset (DMA1_TCD23_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_92E6	TCD Transfer Attributes (DMA1_TCD23_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_92E8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD23_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_92E8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD23_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_92E8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD23_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_92EC	TCD Last Source Address Adjustment (DMA1_TCD23_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_92F0	TCD Destination Address (DMA1_TCD23_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_92F4	TCD Signed Destination Address Offset (DMA1_TCD23_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_92F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD23_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_92F6	DMA1_TCD23_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_92F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD23_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_92FC	TCD Control and Status (DMA1_TCD23_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_92FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD23_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_92FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD23_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9300	TCD Source Address (DMA1_TCD24_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9304	TCD Signed Source Address Offset (DMA1_TCD24_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9306	TCD Transfer Attributes (DMA1_TCD24_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9308	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD24_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9308	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD24_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9308	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD24_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_930C	TCD Last Source Address Adjustment (DMA1_TCD24_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9310	TCD Destination Address (DMA1_TCD24_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9314	TCD Signed Destination Address Offset (DMA1_TCD24_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9316	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD24_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9316	DMA1_TCD24_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9318	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD24_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_931C	TCD Control and Status (DMA1_TCD24_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_931E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD24_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_931E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD24_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9320	TCD Source Address (DMA1_TCD25_SADDR)	32	R/W	Undefined	21.3.18/ 1025

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9324	TCD Signed Source Address Offset (DMA1_TCD25_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9326	TCD Transfer Attributes (DMA1_TCD25_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9328	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD25_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9328	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD25_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9328	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD25_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_932C	TCD Last Source Address Adjustment (DMA1_TCD25_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9330	TCD Destination Address (DMA1_TCD25_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9334	TCD Signed Destination Address Offset (DMA1_TCD25_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9336	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD25_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9336	DMA1_TCD25_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9338	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD25_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_933C	TCD Control and Status (DMA1_TCD25_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_933E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD25_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_933E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD25_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9340	TCD Source Address (DMA1_TCD26_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9344	TCD Signed Source Address Offset (DMA1_TCD26_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9346	TCD Transfer Attributes (DMA1_TCD26_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9348	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD26_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9348	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD26_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9348	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD26_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_934C	TCD Last Source Address Adjustment (DMA1_TCD26_SLAST)	32	R/W	Undefined	21.3.24/ 1030

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_9350	TCD Destination Address (DMA1_TCD26_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9354	TCD Signed Destination Address Offset (DMA1_TCD26_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9356	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD26_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9356	DMA1_TCD26_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9358	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD26_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_935C	TCD Control and Status (DMA1_TCD26_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_935E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD26_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_935E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD26_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9360	TCD Source Address (DMA1_TCD27_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9364	TCD Signed Source Address Offset (DMA1_TCD27_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9366	TCD Transfer Attributes (DMA1_TCD27_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9368	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD27_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9368	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD27_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9368	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD27_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_936C	TCD Last Source Address Adjustment (DMA1_TCD27_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9370	TCD Destination Address (DMA1_TCD27_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9374	TCD Signed Destination Address Offset (DMA1_TCD27_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9376	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD27_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9376	DMA1_TCD27_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9378	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD27_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_937C	TCD Control and Status (DMA1_TCD27_CSR)	16	R/W	Undefined	21.3.30/ 1034

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_937E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD27_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_937E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD27_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_9380	TCD Source Address (DMA1_TCD28_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_9384	TCD Signed Source Address Offset (DMA1_TCD28_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_9386	TCD Transfer Attributes (DMA1_TCD28_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_9388	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD28_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_9388	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD28_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_9388	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD28_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_938C	TCD Last Source Address Adjustment (DMA1_TCD28_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_9390	TCD Destination Address (DMA1_TCD28_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_9394	TCD Signed Destination Address Offset (DMA1_TCD28_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_9396	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD28_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_9396	DMA1_TCD28_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_9398	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD28_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_939C	TCD Control and Status (DMA1_TCD28_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_939E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD28_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_939E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD28_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_93A0	TCD Source Address (DMA1_TCD29_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_93A4	TCD Signed Source Address Offset (DMA1_TCD29_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_93A6	TCD Transfer Attributes (DMA1_TCD29_ATTR)	16	R/W	Undefined	21.3.20/ 1026

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_93A8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD29_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_93A8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD29_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_93A8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD29_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_93AC	TCD Last Source Address Adjustment (DMA1_TCD29_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_93B0	TCD Destination Address (DMA1_TCD29_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_93B4	TCD Signed Destination Address Offset (DMA1_TCD29_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_93B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD29_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_93B6	DMA1_TCD29_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_93B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD29_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_93BC	TCD Control and Status (DMA1_TCD29_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_93BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD29_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_93BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD29_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_93C0	TCD Source Address (DMA1_TCD30_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_93C4	TCD Signed Source Address Offset (DMA1_TCD30_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_93C6	TCD Transfer Attributes (DMA1_TCD30_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_93C8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD30_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_93C8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD30_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_93C8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD30_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_93CC	TCD Last Source Address Adjustment (DMA1_TCD30_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_93D0	TCD Destination Address (DMA1_TCD30_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_93D4	TCD Signed Destination Address Offset (DMA1_TCD30_DOFF)	16	R/W	Undefined	21.3.26/ 1031

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_93D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD30_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_93D6	DMA1_TCD30_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_93D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD30_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_93DC	TCD Control and Status (DMA1_TCD30_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_93DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD30_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037
4009_93DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD30_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038
4009_93E0	TCD Source Address (DMA1_TCD31_SADDR)	32	R/W	Undefined	21.3.18/ 1025
4009_93E4	TCD Signed Source Address Offset (DMA1_TCD31_SOFF)	16	R/W	Undefined	21.3.19/ 1025
4009_93E6	TCD Transfer Attributes (DMA1_TCD31_ATTR)	16	R/W	Undefined	21.3.20/ 1026
4009_93E8	TCD Minor Byte Count (Minor Loop Disabled) (DMA1_TCD31_NBYTES_MLNO)	32	R/W	Undefined	21.3.21/ 1027
4009_93E8	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA1_TCD31_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.22/ 1027
4009_93E8	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA1_TCD31_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.23/ 1029
4009_93EC	TCD Last Source Address Adjustment (DMA1_TCD31_SLAST)	32	R/W	Undefined	21.3.24/ 1030
4009_93F0	TCD Destination Address (DMA1_TCD31_DADDR)	32	R/W	Undefined	21.3.25/ 1030
4009_93F4	TCD Signed Destination Address Offset (DMA1_TCD31_DOFF)	16	R/W	Undefined	21.3.26/ 1031
4009_93F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD31_CITER_ELINKYES)	16	R/W	Undefined	21.3.27/ 1031
4009_93F6	DMA1_TCD31_CITER_ELINKNO	16	R/W	Undefined	21.3.28/ 1033
4009_93F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA1_TCD31_DLASTSGA)	32	R/W	Undefined	21.3.29/ 1034
4009_93FC	TCD Control and Status (DMA1_TCD31_CSR)	16	R/W	Undefined	21.3.30/ 1034
4009_93FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA1_TCD31_BITER_ELINKYES)	16	R/W	Undefined	21.3.31/ 1037

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4009_93FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA1_TCD31_BITER_ELINKNO)	16	R/W	Undefined	21.3.32/ 1038

21.3.1 Control Register (DMAx_CR)

The CR defines the basic operating configuration of the DMA. The DMA arbitrates channel service requests in two groups of 16 channels each:

- Group 1 contains channels 31-16
- Group 0 contains channels 15-0

Arbitration within a group can be configured to use either a fixed-priority or a round-robin scheme. For fixed-priority arbitration, the highest priority channel requesting service is selected to execute. The channel priority registers assign the priorities; see the DCHPRIn registers. For round-robin arbitration, the channel priorities are ignored and channels within each group are cycled through (from high to low channel number) without regard to priority.

NOTE

For proper operation, writes to the CR register must be performed only when the DMA channels are inactive; that is, when TCDn_CSR[ACTIVE] bits are cleared.

The group priorities operate in a similar fashion. In group fixed priority arbitration mode, channel service requests in the highest priority group are executed first, where priority level 1 is the highest and priority level 0 is the lowest. The group priorities are assigned in the GRPnPRI fields of the DMA Control Register (CR). All group priorities must have unique values prior to any channel service requests occurring; otherwise, a configuration error will be reported. For group round robin arbitration, the group priorities are ignored and the groups are cycled through (from high to low group number) without regard to priority.

Minor loop offsets are address offset values added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. When minor loop offsets are enabled, the minor loop offset (MLOFF) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both prior to the addresses being written back into the TCD. If the major loop is

complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (EMLM is 1), TCDn word2 is redefined. A portion of TCDn word2 is used to specify multiple fields: a source enable bit (SMLOE) to specify the minor loop offset should be applied to the source address (TCDn_SADDR) upon minor loop completion, a destination enable bit (DMLOE) to specify the minor loop offset should be applied to the destination address (TCDn_DADDR) upon minor loop completion, and the sign extended minor loop offset value (MLOFF). The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. When both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

When minor loop mapping is disabled (EMLM is 0), all 32 bits of TCDn word2 are assigned to the NBYTES field.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														CX	ECX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0	GRP1PRI	0	GRP0PRI	EMLM	CLM	HALT	HOE	ERGA	ERCA	EDBG	0
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

DMAx_CR field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 CX	Cancel Transfer 0 Normal operation 1 Cancel the remaining data transfer. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The CX bit clears itself after the cancel has been honored. This cancel retires the channel normally as if the minor loop was completed.
16 ECX	Error Cancel Transfer

Table continues on the next page...

DMAx_CR field descriptions (continued)

Field	Description
	0 Normal operation 1 Cancel the remaining data transfer in the same fashion as the CX bit. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The ECX bit clears itself after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating the Error Status register (DMAx_ES) and generating an optional error interrupt.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 GRP1PRI	Channel Group 1 Priority Group 1 priority level when fixed priority group arbitration is enabled.
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 GRP0PRI	Channel Group 0 Priority Group 0 priority level when fixed priority group arbitration is enabled.
7 EMLM	Enable Minor Loop Mapping 0 Disabled. TCDn.word2 is defined as a 32-bit NBYTES field. 1 Enabled. TCDn.word2 is redefined to include individual enable fields, an offset field, and the NBYTES field. The individual enable fields allow the minor loop offset to be applied to the source address, the destination address, or both. The NBYTES field is reduced when either offset is enabled.
6 CLM	Continuous Link Mode 0 A minor loop channel link made to itself goes through channel arbitration before being activated again. 1 A minor loop channel link made to itself does not go through channel arbitration before being activated again. Upon minor loop completion, the channel activates again if that channel has a minor loop channel link enabled and the link channel is itself. This effectively applies the minor loop offsets and restarts the next minor loop.
5 HALT	Halt DMA Operations 0 Normal operation 1 Stall the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this bit is cleared.
4 HOE	Halt On Error 0 Normal operation 1 Any error causes the HALT bit to set. Subsequently, all service requests are ignored until the HALT bit is cleared.
3 ERGA	Enable Round Robin Group Arbitration 0 Fixed priority arbitration is used for selection among the groups. 1 Round robin arbitration is used for selection among the groups.
2 ERCA	Enable Round Robin Channel Arbitration 0 Fixed priority arbitration is used for channel selection within each group. 1 Round robin arbitration is used for channel selection within each group.

Table continues on the next page...

DMAx_CR field descriptions (continued)

Field	Description
1 EDBG	Enable Debug 0 When in debug mode, the DMA continues to operate. 1 When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. Channel execution resumes when the system exits debug mode or the EDBG bit is cleared.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

21.3.2 Error Status Register (DMAx_ES)

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- A configuration error, that is:
  - An illegal setting in the transfer-control descriptor, or
  - An illegal priority register setting in fixed-arbitration
- An error termination to a bus master read or write cycle

See the Error Reporting and Handling section for more details.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	0														ECX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPE	CPE	0	ERRCHN					SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMAx_ES field descriptions

Field	Description
31 VLD	Logical OR of all ERR status bits 0 No ERR bits are set 1 At least one ERR bit is set indicating a valid error exists that has not been cleared
30–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 ECX	Transfer Canceled

Table continues on the next page...

DMAx_ES field descriptions (continued)

Field	Description
	0 No canceled transfers 1 The last recorded entry was a canceled transfer by the error cancel transfer input
15 GPE	Group Priority Error 0 No group priority error 1 The last recorded error was a configuration error among the group priorities. All group priorities are not unique.
14 CPE	Channel Priority Error 0 No channel priority error 1 The last recorded error was a configuration error in the channel priorities within a group. Channel priorities within a group are not unique.
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error, excluding GPE and CPE errors, or last recorded error canceled transfer.
7 SAE	Source Address Error 0 No source address configuration error. 1 The last recorded error was a configuration error detected in the TCDn_SADDR field. TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].
6 SOE	Source Offset Error 0 No source offset configuration error 1 The last recorded error was a configuration error detected in the TCDn_SOFF field. TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE].
5 DAE	Destination Address Error 0 No destination address configuration error 1 The last recorded error was a configuration error detected in the TCDn_DADDR field. TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE].
4 DOE	Destination Offset Error 0 No destination offset configuration error 1 The last recorded error was a configuration error detected in the TCDn_DOFF field. TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE].
3 NCE	NBYTES/CITER Configuration Error 0 No NBYTES/CITER configuration error 1 The last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields. • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE], or • TCDn_CITER[CITER] is equal to zero, or • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK]
2 SGE	Scatter/Gather Configuration Error

Table continues on the next page...

DMAx_ES field descriptions (continued)

Field	Description
	0 No scatter/gather configuration error 1 The last recorded error was a configuration error detected in the TCDn_DLASTSGA field. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. TCDn_DLASTSGA is not on a 32 byte boundary.
1 SBE	Source Bus Error 0 No source bus error 1 The last recorded error was a bus error on a source read
0 DBE	Destination Bus Error 0 No destination bus error 1 The last recorded error was a bus error on a destination write

21.3.3 Enable Request Register (DMAx_ERQ)

The ERQ register provides a bit map for the 32 implemented channels to enable the request signal for each channel. The state of any given channel enable is directly affected by writes to this register; it is also affected by writes to the SERQ and CERQ. The {S,C}ERQ registers are provided so the request enable for a single channel can easily be modified without needing to perform a read-modify-write sequence to the ERQ.

DMA request input signals and this enable request flag must be asserted before a channel's hardware service request is accepted. The state of the DMA enable request flag does not affect a channel service request made explicitly through software or a linked channel request.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	ERQ31	ERQ30	ERQ29	ERQ28	ERQ27	ERQ26	ERQ25	ERQ24	ERQ23	ERQ22	ERQ21	ERQ20	ERQ19	ERQ18	ERQ17	ERQ16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ERQ15	ERQ14	ERQ13	ERQ12	ERQ11	ERQ10	ERQ9	ERQ8	ERQ7	ERQ6	ERQ5	ERQ4	ERQ3	ERQ2	ERQ1	ERQ0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMAx_ERQ field descriptions

Field	Description
31 ERQ31	Enable DMA Request 31 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
30 ERQ30	Enable DMA Request 30 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
29 ERQ29	Enable DMA Request 29 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
28 ERQ28	Enable DMA Request 28 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
27 ERQ27	Enable DMA Request 27 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
26 ERQ26	Enable DMA Request 26 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
25 ERQ25	Enable DMA Request 25 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
24 ERQ24	Enable DMA Request 24 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
23 ERQ23	Enable DMA Request 23 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
22 ERQ22	Enable DMA Request 22 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
21 ERQ21	Enable DMA Request 21 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
20 ERQ20	Enable DMA Request 20 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled

Table continues on the next page...

DMAx_ERQ field descriptions (continued)

Field	Description
19 ERQ19	Enable DMA Request 19 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
18 ERQ18	Enable DMA Request 18 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
17 ERQ17	Enable DMA Request 17 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
16 ERQ16	Enable DMA Request 16 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
15 ERQ15	Enable DMA Request 15 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
14 ERQ14	Enable DMA Request 14 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
13 ERQ13	Enable DMA Request 13 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
12 ERQ12	Enable DMA Request 12 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
11 ERQ11	Enable DMA Request 11 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
10 ERQ10	Enable DMA Request 10 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
9 ERQ9	Enable DMA Request 9 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
8 ERQ8	Enable DMA Request 8 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled

Table continues on the next page...

DMAx_ERQ field descriptions (continued)

Field	Description
7 ERQ7	Enable DMA Request 7 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
6 ERQ6	Enable DMA Request 6 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
5 ERQ5	Enable DMA Request 5 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
4 ERQ4	Enable DMA Request 4 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
3 ERQ3	Enable DMA Request 3 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
2 ERQ2	Enable DMA Request 2 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
1 ERQ1	Enable DMA Request 1 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
0 ERQ0	Enable DMA Request 0 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled

21.3.4 Enable Error Interrupt Register (DMAx_EEI)

The EEI register provides a bit map for the 32 channels to enable the error interrupt signal for each channel. The state of any given channel's error interrupt enable is directly affected by writes to this register; it is also affected by writes to the SEEI and CEEI. The {S,C}EEI are provided so the error interrupt enable for a single channel can easily be modified without the need to perform a read-modify-write sequence to the EEI register.

The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	EEI31	EEI30	EEI29	EEI28	EEI27	EEI26	EEI25	EEI24	EEI23	EEI22	EEI21	EEI20	EEI19	EEI18	EEI17	EEI16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	EEI15	EEI14	EEI13	EEI12	EEI11	EEI10	EEI9	EEI8	EEI7	EEI6	EEI5	EEI4	EEI3	EEI2	EEI1	EEI0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMAx_EEI field descriptions

Field	Description
31 EEI31	Enable Error Interrupt 31 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
30 EEI30	Enable Error Interrupt 30 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
29 EEI29	Enable Error Interrupt 29 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
28 EEI28	Enable Error Interrupt 28 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
27 EEI27	Enable Error Interrupt 27 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
26 EEI26	Enable Error Interrupt 26 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
25 EEI25	Enable Error Interrupt 25 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
24 EEI24	Enable Error Interrupt 24 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request

Table continues on the next page...

DMAx_EEI field descriptions (continued)

Field	Description
23 EEI23	Enable Error Interrupt 23 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
22 EEI22	Enable Error Interrupt 22 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
21 EEI21	Enable Error Interrupt 21 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
20 EEI20	Enable Error Interrupt 20 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
19 EEI19	Enable Error Interrupt 19 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
18 EEI18	Enable Error Interrupt 18 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
17 EEI17	Enable Error Interrupt 17 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
16 EEI16	Enable Error Interrupt 16 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
15 EEI15	Enable Error Interrupt 15 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
14 EEI14	Enable Error Interrupt 14 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
13 EEI13	Enable Error Interrupt 13 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
12 EEI12	Enable Error Interrupt 12 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request

Table continues on the next page...

DMAx_EEI field descriptions (continued)

Field	Description
11 EEI11	Enable Error Interrupt 11 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
10 EEI10	Enable Error Interrupt 10 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
9 EEI9	Enable Error Interrupt 9 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
8 EEI8	Enable Error Interrupt 8 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
7 EEI7	Enable Error Interrupt 7 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
6 EEI6	Enable Error Interrupt 6 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
5 EEI5	Enable Error Interrupt 5 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
4 EEI4	Enable Error Interrupt 4 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
3 EEI3	Enable Error Interrupt 3 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
2 EEI2	Enable Error Interrupt 2 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
1 EEI1	Enable Error Interrupt 1 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
0 EEI0	Enable Error Interrupt 0 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request

21.3.5 Clear Enable Error Interrupt Register (DMAx_CEEI)

The CEEI provides a simple memory-mapped mechanism to clear a given bit in the EEI to disable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI to be cleared. Setting the CAEE bit provides a global clear function, forcing the EEI contents to be cleared, disabling all DMA request inputs. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: Base address + 18h offset

Bit	7	6	5	4	3	2	1	0
Read	0	0		0				
Write	NOP	CAEE	0	CEEI				
Reset	0	0	0	0	0	0	0	0

DMAx_CEEI field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CAEE	Clear All Enable Error Interrupts 0 Clear only the EEI bit specified in the CEEI field 1 Clear all bits in EEI
5 Reserved	This field is reserved.
4–0 CEEI	Clear Enable Error Interrupt Clears the corresponding bit in EEI

21.3.6 Set Enable Error Interrupt Register (DMAx_SEEI)

The SEEI provides a simple memory-mapped mechanism to set a given bit in the EEI to enable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI to be set. Setting the SAEE bit provides a global set function, forcing the entire EEI contents to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: Base address + 19h offset

Bit	7	6	5	4	3	2	1	0
Read	0	0		0				
Write	NOP	SAEE	0	SEEI				
Reset	0	0	0	0	0	0	0	0

DMAx_SEEI field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 SAEE	Sets All Enable Error Interrupts 0 Set only the EEI bit specified in the SEEI field. 1 Sets all bits in EEI
5 Reserved	This field is reserved.
4–0 SEEI	Set Enable Error Interrupt Sets the corresponding bit in EEI

21.3.7 Clear Enable Request Register (DMAx_CERQ)

The CERQ provides a simple memory-mapped mechanism to clear a given bit in the ERQ to disable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ to be cleared. Setting the CAER bit provides a global clear function, forcing the entire contents of the ERQ to be cleared, disabling all DMA request inputs. If NOP is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: Base address + 1Ah offset

Bit	7	6	5	4	3	2	1	0
Read	0	0		0				
Write	NOP	CAER	0	CERQ				
Reset	0	0	0	0	0	0	0	0

DMAx_CERQ field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CAER	Clear All Enable Requests 0 Clear only the ERQ bit specified in the CERQ field 1 Clear all bits in ERQ
5 Reserved	This field is reserved.
4–0 CERQ	Clear Enable Request Clears the corresponding bit in ERQ{H,L}

21.3.8 Set Enable Request Register (DMAx_SERQ)

The SERQ provides a simple memory-mapped mechanism to set a given bit in the ERQ to enable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ to be set. Setting the SAER bit provides a global set function, forcing the entire contents of ERQ to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: Base address + 1Bh offset

Bit	7	6	5	4	3	2	1	0
Read	0	0		0				
Write	NOP	SAER	0	SERQ				
Reset	0	0	0	0	0	0	0	0

DMAx_SERQ field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 SAER	Set All Enable Requests 0 Set only the ERQ bit specified in the SERQ field 1 Set all bits in ERQ
5 Reserved	This field is reserved.
4–0 SERQ	Set Enable Request Sets the corresponding bit in ERQ

21.3.9 Clear DONE Status Bit Register (DMAx_CDNE)

The CDNE provides a simple memory-mapped mechanism to clear the DONE bit in the TCD of the given channel. The data value on a register write causes the DONE bit in the corresponding transfer control descriptor to be cleared. Setting the CADN bit provides a global clear function, forcing all DONE bits to be cleared. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: Base address + 1Ch offset

Bit	7	6	5	4	3	2	1	0
Read	0	0		0				
Write	NOP	CADN	0	CDNE				
Reset	0	0	0	0	0	0	0	0

DMAx_CDNE field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CADN	Clears All DONE Bits 0 Clears only the TCDn_CSR[DONE] bit specified in the CDNE field 1 Clears all bits in TCDn_CSR[DONE]
5 Reserved	This field is reserved.
4–0 CDNE	Clear DONE Bit Clears the corresponding bit in TCDn_CSR[DONE]

21.3.10 Set START Bit Register (DMAx_SSRT)

The SSRT provides a simple memory-mapped mechanism to set the START bit in the TCD of the given channel. The data value on a register write causes the START bit in the corresponding transfer control descriptor to be set. Setting the SAST bit provides a global set function, forcing all START bits to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: Base address + 1Dh offset

Bit	7	6	5	4	3	2	1	0
Read	0	0		0				
Write	NOP	SAST	0	SSRT				
Reset	0	0	0	0	0	0	0	0

DMAx_SSRT field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 SAST	Set All START Bits (activates all channels) 0 Set only the TCDn_CSR[START] bit specified in the SSRT field 1 Set all bits in TCDn_CSR[START]
5 Reserved	This field is reserved.
4–0 SSRT	Set START Bit Sets the corresponding bit in TCDn_CSR[START]

21.3.11 Clear Error Register (DMAx_CERR)

The CERR provides a simple memory-mapped mechanism to clear a given bit in the ERR to disable the error condition flag for a given channel. The given value on a register write causes the corresponding bit in the ERR to be cleared. Setting the CAEI bit provides a global clear function, forcing the ERR contents to be cleared, clearing all channel error indicators. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: Base address + 1Eh offset

Bit	7	6	5	4	3	2	1	0
Read	0	0		0				
Write	NOP	CAEI	0	CERR				
Reset	0	0	0	0	0	0	0	0

DMAx_CERR field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CAEI	Clear All Error Indicators 0 Clear only the ERR bit specified in the CERR field 1 Clear all bits in ERR
5 Reserved	This field is reserved.
4–0 CERR	Clear Error Indicator Clears the corresponding bit in ERR

21.3.12 Clear Interrupt Request Register (DMAx_CINT)

The CINT provides a simple, memory-mapped mechanism to clear a given bit in the INT to disable the interrupt request for a given channel. The given value on a register write causes the corresponding bit in the INT to be cleared. Setting the CAIR bit provides a global clear function, forcing the entire contents of the INT to be cleared, disabling all DMA interrupt requests. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: Base address + 1Fh offset

Bit	7	6	5	4	3	2	1	0
Read	0	0		0				
Write	NOP	CAIR	0	CINT				
Reset	0	0	0	0	0	0	0	0

DMAx_CINT field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CAIR	Clear All Interrupt Requests 0 Clear only the INT bit specified in the CINT field 1 Clear all bits in INT
5 Reserved	This field is reserved.
4–0 CINT	Clear interrupt request Clears the corresponding bit in INT

21.3.13 Interrupt Request Register (DMAx_INT)

The INT register provides a bit map for the 32 channels signaling the presence of an interrupt request for each channel. Depending on the appropriate bit setting in the transfer-control descriptors, the eDMA engine generates an interrupt on data transfer completion. The outputs of this register are directly routed to the interrupt controller (INTC). During the interrupt-service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. Typically, a write to the CINT register in the interrupt service routine is used for this purpose.

The state of any given channel's interrupt request is directly affected by writes to this register; it is also affected by writes to the CINT register. On writes to INT, a 1 in any bit position clears the corresponding channel's interrupt request. A zero in any bit position has no affect on the corresponding channel's current interrupt status. The CINT register is provided so the interrupt request for a single channel can easily be cleared without the need to perform a read-modify-write sequence to the INT register.

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMAx_INT field descriptions

Field	Description
31 INT31	Interrupt Request 31 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
30 INT30	Interrupt Request 30 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
29 INT29	Interrupt Request 29 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
28 INT28	Interrupt Request 28 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
27 INT27	Interrupt Request 27

Table continues on the next page...

DMAx_INT field descriptions (continued)

Field	Description
	0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
26 INT26	Interrupt Request 26 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
25 INT25	Interrupt Request 25 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
24 INT24	Interrupt Request 24 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
23 INT23	Interrupt Request 23 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
22 INT22	Interrupt Request 22 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
21 INT21	Interrupt Request 21 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
20 INT20	Interrupt Request 20 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
19 INT19	Interrupt Request 19 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
18 INT18	Interrupt Request 18 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
17 INT17	Interrupt Request 17 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
16 INT16	Interrupt Request 16 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
15 INT15	Interrupt Request 15

Table continues on the next page...

DMAx_INT field descriptions (continued)

Field	Description
	0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
14 INT14	Interrupt Request 14 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
13 INT13	Interrupt Request 13 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
12 INT12	Interrupt Request 12 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
11 INT11	Interrupt Request 11 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
10 INT10	Interrupt Request 10 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
9 INT9	Interrupt Request 9 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
8 INT8	Interrupt Request 8 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
7 INT7	Interrupt Request 7 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
6 INT6	Interrupt Request 6 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
5 INT5	Interrupt Request 5 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
4 INT4	Interrupt Request 4 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
3 INT3	Interrupt Request 3

Table continues on the next page...

DMAx_INT field descriptions (continued)

Field	Description
	0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
2 INT2	Interrupt Request 2 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
1 INT1	Interrupt Request 1 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
0 INT0	Interrupt Request 0 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active

21.3.14 Error Register (DMAx_ERR)

The ERR provides a bit map for the 32 channels, signaling the presence of an error for each channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate bit in this register. The outputs of this register are enabled by the contents of the EEI, then logically summed across groups of 16 and 32 channels to form several group error interrupt requests, which are then routed to the interrupt controller. During the execution of the interrupt-service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. Typically, a write to the CERR in the interrupt-service routine is used for this purpose. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when an error is detected.

The contents of this register can also be polled because a non-zero value indicates the presence of a channel error regardless of the state of the EEI. The state of any given channel's error indicators is affected by writes to this register; it is also affected by writes to the CERR. On writes to the ERR, a one in any bit position clears the corresponding channel's error status. A zero in any bit position has no affect on the corresponding channel's current error status. The CERR is provided so the error indicator for a single channel can easily be cleared.

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR31	ERR30	ERR29	ERR28	ERR27	ERR26	ERR25	ERR24	ERR23	ERR22	ERR21	ERR20	ERR19	ERR18	ERR17	ERR16
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR15	ERR14	ERR13	ERR12	ERR11	ERR10	ERR9	ERR8	ERR7	ERR6	ERR5	ERR4	ERR3	ERR2	ERR1	ERR0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMAx_ERR field descriptions

Field	Description
31 ERR31	Error In Channel 31 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
30 ERR30	Error In Channel 30 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
29 ERR29	Error In Channel 29 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
28 ERR28	Error In Channel 28 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
27 ERR27	Error In Channel 27 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
26 ERR26	Error In Channel 26 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
25 ERR25	Error In Channel 25

Table continues on the next page...

DMAx_ERR field descriptions (continued)

Field	Description
	0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
24 ERR24	Error In Channel 24 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
23 ERR23	Error In Channel 23 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
22 ERR22	Error In Channel 22 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
21 ERR21	Error In Channel 21 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
20 ERR20	Error In Channel 20 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
19 ERR19	Error In Channel 19 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
18 ERR18	Error In Channel 18 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
17 ERR17	Error In Channel 17 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
16 ERR16	Error In Channel 16 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
15 ERR15	Error In Channel 15 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
14 ERR14	Error In Channel 14 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
13 ERR13	Error In Channel 13

Table continues on the next page...

DMAx_ERR field descriptions (continued)

Field	Description
	0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
12 ERR12	Error In Channel 12 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
11 ERR11	Error In Channel 11 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
10 ERR10	Error In Channel 10 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
9 ERR9	Error In Channel 9 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
8 ERR8	Error In Channel 8 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
7 ERR7	Error In Channel 7 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
6 ERR6	Error In Channel 6 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
5 ERR5	Error In Channel 5 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
4 ERR4	Error In Channel 4 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
3 ERR3	Error In Channel 3 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
2 ERR2	Error In Channel 2 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
1 ERR1	Error In Channel 1

Table continues on the next page...

DMAx_ERR field descriptions (continued)

Field	Description
	0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
0 ERR0	Error In Channel 0 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred

21.3.15 Hardware Request Status Register (DMAx_HRS)

The HRS register provides a bit map for the DMA channels, signaling the presence of a hardware request for each channel. The hardware request status bits reflect the current state of the register and qualified (via the ERQ fields) DMA request signals as seen by the DMA's arbitration logic. This view into the hardware request signals may be used for debug purposes.

NOTE

These bits reflect the state of the request as seen by the arbitration logic. Therefore, this status is affected by the ERQ bits.

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	HRS31	HRS30	HRS29	HRS28	HRS27	HRS26	HRS25	HRS24	HRS23	HRS22	HRS21	HRS20	HRS19	HRS18	HRS17	HRS16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	HRS15	HRS14	HRS13	HRS12	HRS11	HRS10	HRS9	HRS8	HRS7	HRS6	HRS5	HRS4	HRS3	HRS2	HRS1	HRS0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMAx_HRS field descriptions

Field	Description
31 HRS31	Hardware Request Status Channel 31 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present

Table continues on the next page...

DMAx_HRS field descriptions (continued)

Field	Description
30 HRS30	Hardware Request Status Channel 30 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
29 HRS29	Hardware Request Status Channel 29 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
28 HRS28	Hardware Request Status Channel 28 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
27 HRS27	Hardware Request Status Channel 27 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
26 HRS26	Hardware Request Status Channel 26 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
25 HRS25	Hardware Request Status Channel 25 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
24 HRS24	Hardware Request Status Channel 24 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
23 HRS23	Hardware Request Status Channel 23 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
22 HRS22	Hardware Request Status Channel 22 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
21 HRS21	Hardware Request Status Channel 21 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
20 HRS20	Hardware Request Status Channel 20 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
19 HRS19	Hardware Request Status Channel 19 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present

Table continues on the next page...

DMAx_HRS field descriptions (continued)

Field	Description
18 HRS18	Hardware Request Status Channel 18 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
17 HRS17	Hardware Request Status Channel 17 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
16 HRS16	Hardware Request Status Channel 16 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
15 HRS15	Hardware Request Status Channel 15 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
14 HRS14	Hardware Request Status Channel 14 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
13 HRS13	Hardware Request Status Channel 13 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
12 HRS12	Hardware Request Status Channel 12 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
11 HRS11	Hardware Request Status Channel 11 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
10 HRS10	Hardware Request Status Channel 10 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
9 HRS9	Hardware Request Status Channel 9 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
8 HRS8	Hardware Request Status Channel 8 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
7 HRS7	Hardware Request Status Channel 7 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present

Table continues on the next page...

DMAx_HRS field descriptions (continued)

Field	Description
6 HRS6	Hardware Request Status Channel 6 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
5 HRS5	Hardware Request Status Channel 5 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
4 HRS4	Hardware Request Status Channel 4 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
3 HRS3	Hardware Request Status Channel 3 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
2 HRS2	Hardware Request Status Channel 2 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
1 HRS1	Hardware Request Status Channel 1 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
0 HRS0	Hardware Request Status Channel 0 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present

21.3.16 Enable Asynchronous Request in Stop Register (DMAx_EARS)

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EDREQ_31	EDREQ_30	EDREQ_29	EDREQ_28	EDREQ_27	EDREQ_26	EDREQ_25	EDREQ_24	EDREQ_23	EDREQ_22	EDREQ_21	EDREQ_20	EDREQ_19	EDREQ_18	EDREQ_17	EDREQ_16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EDREQ_15	EDREQ_14	EDREQ_13	EDREQ_12	EDREQ_11	EDREQ_10	EDREQ_9	EDREQ_8	EDREQ_7	EDREQ_6	EDREQ_5	EDREQ_4	EDREQ_3	EDREQ_2	EDREQ_1	EDREQ_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMAx_EARS field descriptions

Field	Description
31 EDREQ_31	Enable asynchronous DMA request in stop for channel 31 0 Disable asynchronous DMA request for channel 31 1 Enable asynchronous DMA request for channel 31
30 EDREQ_30	Enable asynchronous DMA request in stop for channel 30 0 Disable asynchronous DMA request for channel 30 1 Enable asynchronous DMA request for channel 30
29 EDREQ_29	Enable asynchronous DMA request in stop for channel 29 0 Disable asynchronous DMA request for channel 29 1 Enable asynchronous DMA request for channel 29
28 EDREQ_28	Enable asynchronous DMA request in stop for channel 28 0 Disable asynchronous DMA request for channel 28 1 Enable asynchronous DMA request for channel 28
27 EDREQ_27	Enable asynchronous DMA request in stop for channel 27 0 Disable asynchronous DMA request for channel 27 1 Enable asynchronous DMA request for channel 27
26 EDREQ_26	Enable asynchronous DMA request in stop for channel 26 0 Disable asynchronous DMA request for channel 26 1 Enable asynchronous DMA request for channel 26
25 EDREQ_25	Enable asynchronous DMA request in stop for channel 25 0 Disable asynchronous DMA request for channel 25 1 Enable asynchronous DMA request for channel 25
24 EDREQ_24	Enable asynchronous DMA request in stop for channel 24 0 Disable asynchronous DMA request for channel 24 1 Enable asynchronous DMA request for channel 24
23 EDREQ_23	Enable asynchronous DMA request in stop for channel 23 0 Disable asynchronous DMA request for channel 23 1 Enable asynchronous DMA request for channel 23
22 EDREQ_22	Enable asynchronous DMA request in stop for channel 22 0 Disable asynchronous DMA request for channel 22 1 Enable asynchronous DMA request for channel 22
21 EDREQ_21	Enable asynchronous DMA request in stop for channel 21 0 Disable asynchronous DMA request for channel 21 1 Enable asynchronous DMA request for channel 21
20 EDREQ_20	Enable asynchronous DMA request in stop for channel 20 0 Disable asynchronous DMA request for channel 20 1 Enable asynchronous DMA request for channel 20

Table continues on the next page...

DMAx_EARS field descriptions (continued)

Field	Description
19 EDREQ_19	Enable asynchronous DMA request in stop for channel 19 0 Disable asynchronous DMA request for channel 19 1 Enable asynchronous DMA request for channel 19
18 EDREQ_18	Enable asynchronous DMA request in stop for channel 18 0 Disable asynchronous DMA request for channel 18 1 Enable asynchronous DMA request for channel 18
17 EDREQ_17	Enable asynchronous DMA request in stop for channel 17 0 Disable asynchronous DMA request for channel 17 1 Enable asynchronous DMA request for channel 17
16 EDREQ_16	Enable asynchronous DMA request in stop for channel 16 0 Disable asynchronous DMA request for channel 16 1 Enable asynchronous DMA request for channel 16
15 EDREQ_15	Enable asynchronous DMA request in stop for channel 15 0 Disable asynchronous DMA request for channel 15. 1 Enable asynchronous DMA request for channel 15.
14 EDREQ_14	Enable asynchronous DMA request in stop for channel 14 0 Disable asynchronous DMA request for channel 14. 1 Enable asynchronous DMA request for channel 14.
13 EDREQ_13	Enable asynchronous DMA request in stop for channel 13 0 Disable asynchronous DMA request for channel 13. 1 Enable asynchronous DMA request for channel 13.
12 EDREQ_12	Enable asynchronous DMA request in stop for channel 12 0 Disable asynchronous DMA request for channel 12. 1 Enable asynchronous DMA request for channel 12.
11 EDREQ_11	Enable asynchronous DMA request in stop for channel 11 0 Disable asynchronous DMA request for channel 11. 1 Enable asynchronous DMA request for channel 11.
10 EDREQ_10	Enable asynchronous DMA request in stop for channel 10 0 Disable asynchronous DMA request for channel 10. 1 Enable asynchronous DMA request for channel 10.
9 EDREQ_9	Enable asynchronous DMA request in stop for channel 9 0 Disable asynchronous DMA request for channel 9. 1 Enable asynchronous DMA request for channel 9.
8 EDREQ_8	Enable asynchronous DMA request in stop for channel 8 0 Disable asynchronous DMA request for channel 8. 1 Enable asynchronous DMA request for channel 8.

Table continues on the next page...

DMAx_EARS field descriptions (continued)

Field	Description
7 EDREQ_7	Enable asynchronous DMA request in stop for channel 7 0 Disable asynchronous DMA request for channel 7. 1 Enable asynchronous DMA request for channel 7.
6 EDREQ_6	Enable asynchronous DMA request in stop for channel 6 0 Disable asynchronous DMA request for channel 6. 1 Enable asynchronous DMA request for channel 6.
5 EDREQ_5	Enable asynchronous DMA request in stop for channel 5 0 Disable asynchronous DMA request for channel 5. 1 Enable asynchronous DMA request for channel 5.
4 EDREQ_4	Enable asynchronous DMA request in stop for channel 4 0 Disable asynchronous DMA request for channel 4. 1 Enable asynchronous DMA request for channel 4.
3 EDREQ_3	Enable asynchronous DMA request in stop for channel 3. 0 Disable asynchronous DMA request for channel 3. 1 Enable asynchronous DMA request for channel 3.
2 EDREQ_2	Enable asynchronous DMA request in stop for channel 2. 0 Disable asynchronous DMA request for channel 2. 1 Enable asynchronous DMA request for channel 2.
1 EDREQ_1	Enable asynchronous DMA request in stop for channel 1. 0 Disable asynchronous DMA request for channel 1 1 Enable asynchronous DMA request for channel 1.
0 EDREQ_0	Enable asynchronous DMA request in stop for channel 0. 0 Disable asynchronous DMA request for channel 0. 1 Enable asynchronous DMA request for channel 0.

21.3.17 Channel n Priority Register (DMA*x*_DCHPRI*n*)

When fixed-priority channel arbitration is enabled (CR[ERCA] = 0), the contents of these registers define the unique priorities associated with each channel within a group. The channel priorities are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, etc. Software must program the channel priorities with unique values; otherwise, a configuration error is reported. The range of the priority value is limited to the values of 0 through 15. When read, the GRPPRI bits of the DCHPRIn register reflect the current priority level of the group of channels in which the corresponding channel resides. GRPPRI bits are not affected by writes to the DCHPRIn registers. The group priority is assigned in the DMA control register.

Address: Base address + 100h offset + (1d × i), where i=0d to 31d

Bit	7	6	5	4	3	2	1	0
Read	ECP	DPA	GRPPRI		CHPRI			
Write								
Reset	0	0	*	*	*	*	*	*

* Notes:
- GRPPRI field: See bit field description
- CHPRI field: See bit field description

DMA*x*_DCHPRI*n* field descriptions

Field	Description
7 ECP	Enable Channel Preemption. This bit resets to zero. 0 Channel n cannot be suspended by a higher priority channel's service request 1 Channel n can be temporarily suspended by the service request of a higher priority channel
6 DPA	Disable preempt ability. This bit resets to zero. 0 Channel n can suspend a lower priority channel 1 Channel n cannot suspend any channel, regardless of channel priority
5–4 GRPPRI	Channel n Current Group Priority Group priority assigned to this channel group when fixed-priority arbitration is enabled. These two bits are read only; writes are ignored. **NOTE:** Reset value for the group and channel priority fields, GRPPRI and CHPRI, is equal to the corresponding channel number for each priority register, i.e., DCHPRI31[GRPPRI] = 0b01 and DCHPRI31[CHPRI] equals 0b1111.
3–0 CHPRI	Channel n Arbitration Priority Channel priority when fixed-priority arbitration is enabled

Table continues on the next page...

DMAx_DCHPRIn field descriptions (continued)

Field	Description
	NOTE: Reset value for the group and channel priority fields, GRPPRI and CHPRI, is equal to the corresponding channel number for each priority register, i.e., DCHPRI31[GRPPRI] = 0b01 and DCHPRI31[CHPRI] equals 0b01111.

21.3.18 TCD Source Address (DMAx_TCDn_SADDR)

Address: Base address + 1000h offset + (32d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADDR																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_SADDR field descriptions

Field	Description
31–0 SADDR	Source Address Memory address pointing to the source data.

21.3.19 TCD Signed Source Address Offset (DMAx_TCDn_SOFF)

Address: Base address + 1004h offset + (32d × i), where i=0d to 31d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SOFF															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_SOFF field descriptions

Field	Description
15–0 SOFF	Source Address Signed Offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

21.3.20 TCD Transfer Attributes (DMAx_TCDn_ATTR)

Address: Base address + 1006h offset + (32d × i), where i=0d to 31d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SMOD					SSIZE			DMOD					DSIZE		
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_ATTR field descriptions

Field	Description
15–11 SMOD	Source Address Modulo 0 Source address modulo feature is disabled ≠0 This value defines a specific address range specified to be the value after SADDR + SOFF calculation is performed on the original register value. Setting this field provides the ability to implement a circular data queue easily. For data queues requiring power-of-2 size bytes, the queue should start at a 0-modulo-size address and the SMOD field should be set to the appropriate value for the queue, freezing the desired number of upper address bits. The value programmed into this field specifies the number of lower address bits allowed to change. For a circular queue application, the SOFF is typically set to the transfer size to implement post-increment addressing with the SMOD function constraining the addresses to a 0-modulo-size range.
10–8 SSIZE	Source data transfer size Using a reserved encoding causes a configuration error. 000 8-bit 001 16-bit 010 32-bit 011 64-bit 100 Reserved 101 32-byte 110 Reserved 111 Reserved
7–3 DMOD	Destination Address Modulo See the SMOD definition
2–0 DSIZE	Destination Data Transfer Size See the SSIZE definition

21.3.21 TCD Minor Byte Count (Minor Loop Disabled) (DMAx_TCD*n*_NBYTES_MLNO)

This register, or one of the next two registers (TCD_NBYTES_MLOFFNO, TCD_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is disabled (CR[EMLM] = 0)

If minor loop mapping is enabled, see the TCD_NBYTES_MLOFFNO and TCD_NBYTES_MLOFFYES register descriptions for TCD word 2's definition.

Address: Base address + 1008h offset + (32d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NBYTES																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMAx_TCD*n*_NBYTES_MLNO field descriptions

Field	Description
31–0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. (Although, it may be stalled by using the bandwidth control field, or via preemption.) After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed. **NOTE:** An NBYTES value of 0x0000_0000 is interpreted as a 4 GB transfer.

21.3.22 TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMAx_TCD*n*_NBYTES_MLOFFNO)

One of three registers (this register, TCD_NBYTES_MLNO, or TCD_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- SMLOE = 0 and DMLOE = 0

If minor loop mapping is enabled and SMLOE or DMLOE is set, then refer to the TCD_NBYTES_MLOFFYES register description. If minor loop mapping is disabled, then refer to the TCD_NBYTES_MLNO register description.

Address: Base address + 1008h offset + (32d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	SMLOE	DMLOE	NBYTES_OE													
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	NBYTES_OE															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMAx_TCDn_NBYTES_MLOFFNO field descriptions

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0 The minor loop offset is not applied to the SADDR 1 The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0 The minor loop offset is not applied to the DADDR 1 The minor loop offset is applied to the DADDR
29–0 NBYTES_OE	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. (Although, it may be stalled by using the bandwidth control field, or via preemption.) After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

21.3.23 TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA*x*_TCD*n*_NBYTES_MLOFFYES)

One of three registers (this register, TCD_NBYTES_MLNO, or TCD_NBYTES_MLOFFNO), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- Minor loop offset is enabled (SMLOE or DMLOE = 1)

If minor loop mapping is enabled and SMLOE and DMLOE are cleared, then refer to the TCD_NBYTES_MLOFFNO register description. If minor loop mapping is disabled, then refer to the TCD_NBYTES_MLNO register description.

Address: Base address + 1008h offset + (32d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	SMLOE	DMLOE	MLOFF													
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	MLOFF						NBYTES_OD									
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCD*n*_NBYTES_MLOFFYES field descriptions

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0 The minor loop offset is not applied to the SADDR 1 The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion.

Table continues on the next page...

DMAx_TCDn_NBYTES_MLOFFYES field descriptions (continued)

Field	Description
	0 The minor loop offset is not applied to the DADDR 1 The minor loop offset is applied to the DADDR
29–10 MLOFF	If SMLOE or DMLOE is set, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9–0 NBYTES_OD	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. (Although, it may be stalled by using the bandwidth control field, or via preemption.) After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

21.3.24 TCD Last Source Address Adjustment (DMAx_TCDn_SLAST)

Address: Base address + 100Ch offset + (32d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SLAST																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_SLAST field descriptions

Field	Description
31–0 SLAST	Last Source Address Adjustment Adjustment value added to the source address at the completion of the major iteration count. This value can be applied to restore the source address to the initial value, or adjust the address to reference the next data structure. This register uses two's complement notation; the overflow bit is discarded.

21.3.25 TCD Destination Address (DMAx_TCDn_DADDR)

Address: Base address + 1010h offset + (32d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DADDR																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_DADDR field descriptions

Field	Description
31–0 DADDR	Destination Address Memory address pointing to the destination data.

21.3.26 TCD Signed Destination Address Offset (DMAx_TCDn_DOFF)

Address: Base address + 1014h offset + (32d × i), where i=0d to 31d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	DOFF															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_DOFF field descriptions

Field	Description
15–0 DOFF	Destination Address Signed Offset Sign-extended offset applied to the current destination address to form the next-state value as each destination write is completed.

21.3.27 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMAx_TCDn_CITER_ELINKYES)

If TCDn_CITER[ELINK] is set, the TCDn_CITER register is defined as follows.

Address: Base address + 1016h offset + (32d × i), where i=0d to 31d

Bit	15	14	13	12	11	10	9	8
Read Write	ELINK	0	LINKCH					CITER_LE
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read Write	CITER_LE							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_CITER_ELINKYES field descriptions

Field	Description
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This bit must be equal to the BITER[ELINK] bit; otherwise, a configuration error is reported. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
14 Reserved	This field is reserved.
13–9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request to the channel defined by these five bits by setting that channel's TCDn_CSR[START] bit.
8–0 CITER_LE	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations (e.g., final source and destination address calculations), optionally generating an interrupt to signal channel completion before reloading the CITER field from the beginning iteration count (BITER) field. **NOTE:** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

21.3.28 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMAx_TCDn_CITER_ELINKNO)

If TCDn_CITER[ELINK] is cleared, the TCDn_CITER register is defined as follows.

Address: Base address + 1016h offset + (32d × i), where i=0d to 31d

Bit	15	14	13	12	11	10	9	8
Read Write	ELINK	CITER						
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read Write	CITER							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_CITER_ELINKNO field descriptions

Field	Description
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This bit must be equal to the BITER[ELINK] bit; otherwise, a configuration error is reported. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
14–0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations (e.g., final source and destination address calculations), optionally generating an interrupt to signal channel completion before reloading the CITER field from the beginning iteration count (BITER) field. **NOTE:** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

21.3.29 TCD Last Destination Address Adjustment/Scatter Gather Address (DMAx_TCDn_DLASTSGA)

Address: Base address + 1018h offset + (32d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLASTSGA																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_DLASTSGA field descriptions

Field	Description
31–0 DLASTSGA	Destination last address adjustment or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0), then: • Adjustment value added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo-32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo-32-byte, else a configuration error is reported.

21.3.30 TCD Control and Status (DMAx_TCDn_CSR)

Address: Base address + 101Ch offset + (32d × i), where i=0d to 31d

Bit	15	14	13	12	11	10	9	8
Read	BWC			MAJORLINKCH				
Write			0					
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read Write	DONE	ACTIVE	MAJORELINK	ESG	DREQ	INTHALF	INTMAJOR	START
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA*x*_TCD*n*_CSR field descriptions

Field	Description
15–14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. In general, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces the eDMA to stall after the completion of each read/write access to control the bus request bandwidth seen by the crossbar switch. **NOTE:** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00 No eDMA engine stalls 01 Reserved 10 eDMA engine stalls for 4 cycles after each r/w 11 eDMA engine stalls for 8 cycles after each r/w
13 Reserved	This field is reserved.

12–8
MAJORLINKCH
	Link Channel Number
	
	If (MAJORELINK = 0) then
	• No channel-to-channel linking (or chaining) is performed after the major loop counter is exhausted.
	
	else
	
	• After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by these five bits by setting that channel's TCDn_CSR[START] bit.

7 DONE	Channel Done This flag indicates the eDMA has completed the major loop. The eDMA engine sets it as the CITER count reaches zero; The software clears it, or the hardware when the channel is activated. **NOTE:** This bit must be cleared to write the MAJORELINK or ESG bits. This bit resets to zero.
6 ACTIVE	Channel Active This flag signals the channel is currently in execution. It is set when channel service begins, and the eDMA clears it as the minor loop completes or if any error condition is detected.
5 MAJORELINK	Enable channel-to-channel linking on major loop complete As the channel completes the major loop, this flag enables the linking to another channel, defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. **NOTE:** To support the dynamic linking coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses DLASTSGA as a memory pointer to a 0-modulo-32 address containing a 32-byte data structure loaded as the transfer control descriptor into the local memory.

Table continues on the next page...

DMAx_TCDn_CSR field descriptions (continued)

Field	Description
	NOTE: To support the dynamic scatter/gather coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set. 0 The current channel's TCD is normal format. 1 The current channel's TCD specifies a scatter gather format. The DLASTSGA field provides a memory pointer to the next TCD to be loaded into this channel after the major loop completes its execution.
3 DREQ	Disable Request If this flag is set, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches zero. 0 The channel's ERQ bit is not affected 1 The channel's ERQ bit is cleared when the major loop is complete
2 INTHALF	Enable an interrupt when major counter is half complete. If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT register when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER == (BITER >> 1)). This halfway point interrupt request is provided to support double-buffered (aka ping-pong) schemes or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE:** If BITER is set, do not use INTHALF. Use INTMAJOR instead. 0 The half-point interrupt is disabled 1 The half-point interrupt is enabled
1 INTMAJOR	Enable an interrupt when major iteration count completes If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT when the current major iteration count reaches zero. 0 The end-of-major loop interrupt is disabled 1 The end-of-major loop interrupt is enabled
0 START	Channel Start If this flag is set, the channel is requesting service. The eDMA hardware automatically clears this flag after the channel begins execution. This bit resets to zero. 0 The channel is not explicitly started 1 The channel is explicitly started via a software initiated service request

21.3.31 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA*x*_TCD*n*_BITER_ELINKYES)

If the TCDn_BITER[ELINK] bit is set, the TCDn_BITER register is defined as follows.

Address: Base address + 101Eh offset + (32d × i), where i=0d to 31d

Bit	15	14	13	12	11	10	9	8
Read / Write	ELINK	0 (Write)	LINKCH					BITER_LE
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read / Write	BITER_LE							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA*x*_TCD*n*_BITER_ELINKYES field descriptions

Field	Description
15 ELINK	Enables channel-to-channel linking on minor loop complete As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
14 Reserved	This field is reserved.
13–9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by these six bits by setting that channel's TCDn_CSR[START] bit. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field.
8–0 BITER_LE	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.

Table continues on the next page...

DMAx_TCDn_BITER_ELINKYES field descriptions (continued)

Field	Description
	NOTE: When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

21.3.32 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMAx_TCDn_BITER_ELINKNO)

If the TCDn_BITER[ELINK] bit is cleared, the TCDn_BITER register is defined as follows.

Address: Base address + 101Eh offset + (32d × i), where i=0d to 31d

Bit	15	14	13	12	11	10	9	8
Read Write	ELINK	BITER						
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read Write	BITER							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMAx_TCDn_BITER_ELINKNO field descriptions

Field	Description
15 ELINK	Enables channel-to-channel linking on minor loop complete As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
14–0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.

Table continues on the next page...

DMAx_TCDn_BITER_ELINKNO field descriptions (continued)

Field	Description
	NOTE: When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

21.4 Functional description

21.4.1 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

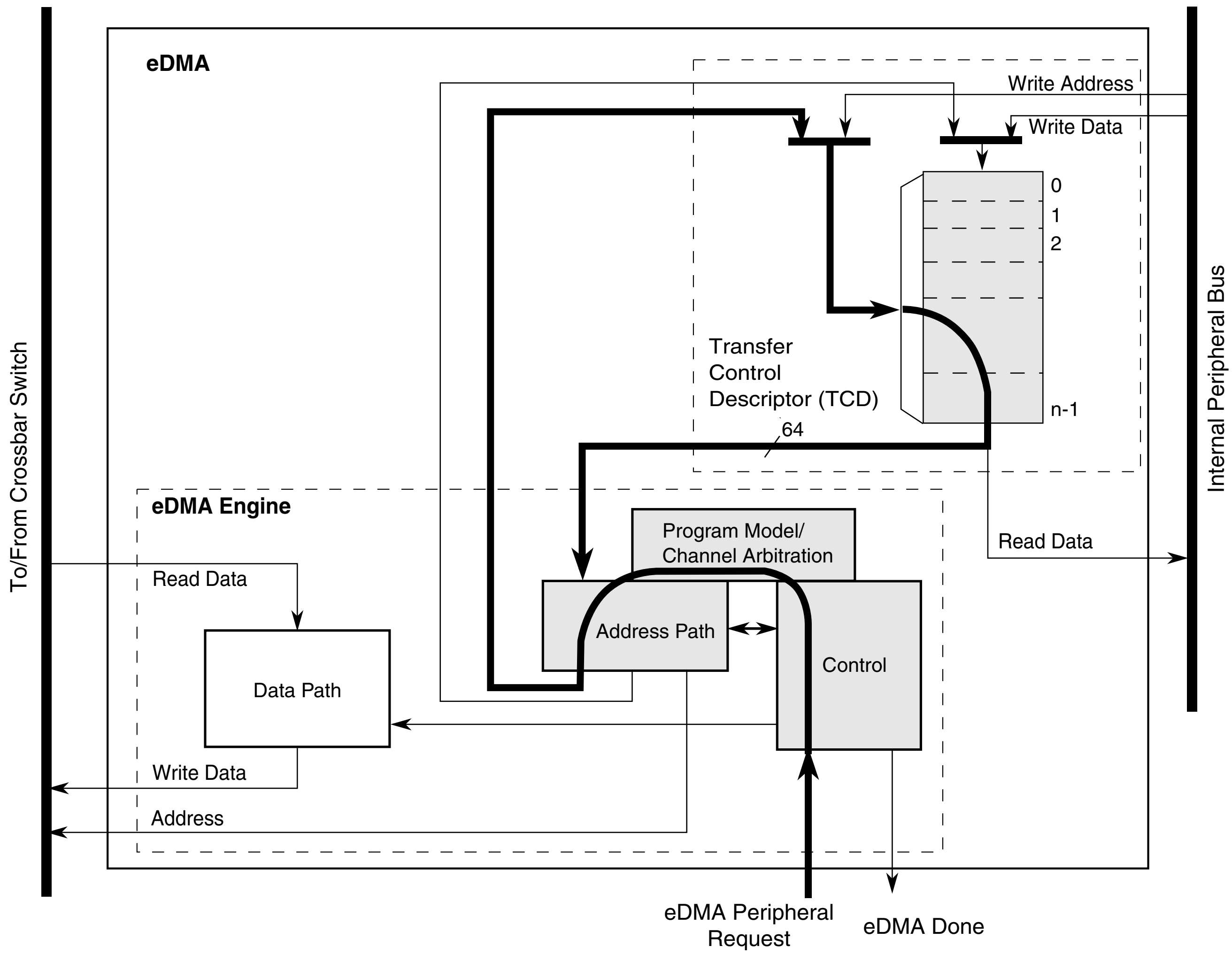


Figure 21-1634. eDMA operation, part 1

This example uses the assertion of the eDMA peripheral request signal to request service for channel *n*. Channel activation via software and the TCD*n*_CSR[START] bit follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration. In the next cycle, the channel arbitration performs, using the fixed-priority or round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for TCD*n*. Next, the TCD memory is accessed and the required descriptor read from the local memory and loaded into the eDMA engine address path channel x or y registers. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path channel x or y registers.

The following diagram illustrates the second part of the basic data flow:

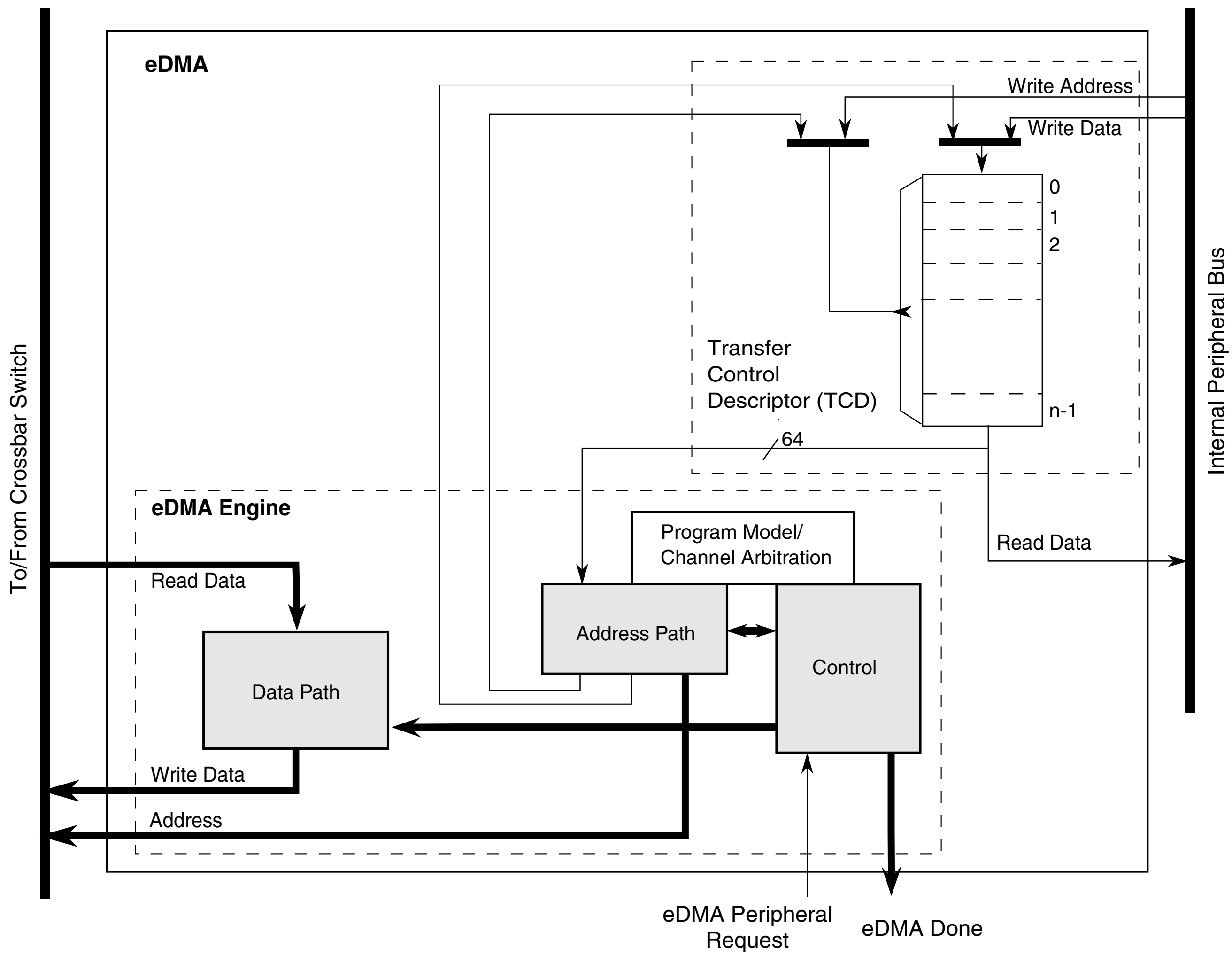


Figure 21-1635. eDMA operation, part 2

The modules associated with the data transfer (address path, data path, and control) sequence through the required source reads and destination writes to perform the actual data movement. The source reads are initiated and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the minor byte count has transferred.

After the minor byte count has moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD, e.g., SADDR, DADDR, CITER. If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of the BITER field into the CITER. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

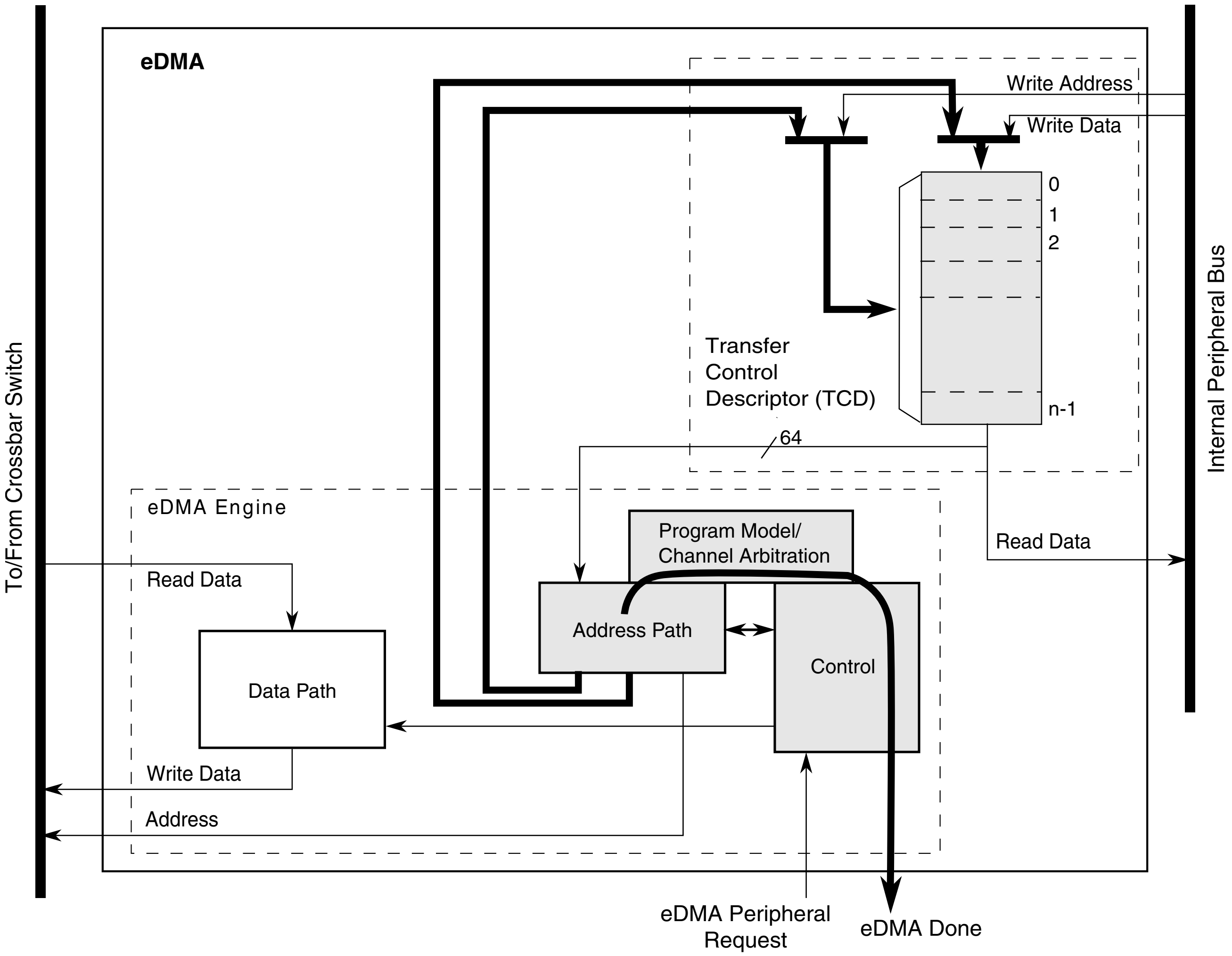


Figure 21-1636. eDMA operation, part 3

21.4.2 Error reporting and handling

Channel errors are reported in the Error Status register (DMAx_ES) and can be caused by:

- A configuration error, which is an illegal setting in the transfer-control descriptor or an illegal priority register setting in Fixed-Arbitration mode, or
- An error termination to a bus master read or write cycle

A configuration error is reported when the starting source or destination address, source or destination offsets, minor loop byte count, or the transfer size represent an inconsistent state. Each of these possible causes are detailed below:

- The addresses and offsets must be aligned on 0-modulo-transfer-size boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.
- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size respectively.
- In fixed arbitration mode, a configuration error is caused by any two channel priorities being equal. All channel priority levels must be unique when fixed arbitration mode is enabled.
- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (DLAST_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn_CITER[E_LINK] bit does not equal the TCDn_BITER[E_LINK] bit.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, report as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion when properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

If a system bus read or write is terminated with an error, the data transfer is stopped and the appropriate bus error flag set. In this case, the state of the channel's transfer control descriptor is updated by the eDMA engine with the current source address, destination address, and current iteration count at the point of the fault. When a system bus error occurs, the channel terminates after the next transfer. Due to pipeline effect, the next transfer is already in progress when the bus error is received by the eDMA. If a bus error occurs on the last read prior to beginning the write sequence, the write executes using the

data captured during the bus error. If a bus error occurs on the last write prior to switching to the next read sequence, the read sequence executes before the channel terminates due to the destination bus error.

A transfer may be cancelled by software with the CR[CX] bit. When a cancel transfer request is recognized, the DMA engine stops processing the channel. The current read-write sequence is allowed to finish. If the cancel occurs on the last read-write sequence of a major or minor loop, the cancel request is discarded and the channel retires normally.

The error cancel transfer is the same as a cancel transfer except the Error Status register (DMAx_ES) is updated with the cancelled channel number and ECX is set. The TCD of a cancelled channel contains the source and destination addresses of the last transfer saved in the TCD. If the channel needs to be restarted, you must re-initialize the TCD because the aforementioned fields no longer represent the original parameters. When a transfer is cancelled by the error cancel transfer mechanism, the channel number is loaded into DMA_ES[ERRCHN] and ECX and VLD are set. In addition, an error interrupt may be generated if enabled.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states and the transaction to the system bus still has peipeline effect), and the appropriate channel bit in the eDMA error register is asserted. At the same time, the details of the error condition are loaded into the Error Status register (DMAx_ES). The major loop complete indicators, setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request, are not affected when an error is detected. After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

21.4.3 Channel preemption

Channel preemption is enabled on a per-channel basis by setting the DCHPRIn[ECP] bit. Channel preemption allows the executing channel’s data transfers to temporarily suspend in favor of starting a higher priority channel. After the preempting channel has completed all its minor loop data transfers, the preempted channel is restored and resumes execution. After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended and the higher priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted. Preemption is available only when fixed arbitration is selected.

A channel's ability to preempt another channel can be disabled by setting DCHPRIn[DPA]. When a channel's preempt ability is disabled, that channel cannot suspend a lower priority channel's data transfer, regardless of the lower priority channel's ECP setting. This allows for a pool of low priority, large data-moving channels to be defined. These low priority channels can be configured to not preempt each other, thus preventing a low priority channel from consuming the preempt slot normally available to a true, high priority channel.

21.4.4 Performance

This section addresses the performance of the eDMA module, focusing on two separate metrics:

- In the traditional data movement context, performance is best expressed as the peak data transfer rates achieved using the eDMA. In most implementations, this transfer rate is limited by the speed of the source and destination address spaces.
- In a second context where device-paced movement of single data values to/from peripherals is dominant, a measure of the requests that can be serviced in a fixed time is a more relevant metric. In this environment, the speed of the source and destination address spaces remains important. However, the microarchitecture of the eDMA also factors significantly into the resulting metric.

21.4.4.1 Peak transfer rates

The peak transfer rates for several different source and destination transfers are shown in the following tables. These tables assume:

- Internal SRAM can be accessed with zero wait-states when viewed from the system bus data phase
- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states, when viewed from the system bus data phase
- All internal peripheral bus accesses are 32-bits in size

This table compares peak transfer rates based n different possible system speeds. Specific chips/devices may not support all system speeds listed.

Table 21-1639. eDMA peak transfer rates (Mbytes/sec)

System Speed, Width	Internal SRAM-to-Internal SRAM	32b internal peripheral bus-to-Internal SRAM	Internal SRAM-to-32b internal peripheral bus
66.7 MHz, 64b	266.7	66.6	53.3
83.3 MHz, 64b	333.3	83.3	66.7
100.0 MHz, 64b	400.0	100.0	80.0
133.3 MHz, 64b	533.3	133.3	106.7
150.0 MHz, 64b	600.0	150.0	120.0

Internal-SRAM-to-internal-SRAM transfers occur at the core's datapath width. For all transfers involving the internal peripheral bus, 32-bit transfer sizes are used. In all cases, the transfer rate includes the time to read the source plus the time to write the destination.

21.4.4.2 Peak request rates

The second performance metric is a measure of the number of DMA requests that can be serviced in a given amount of time. For this metric, assume that the peripheral request causes the channel to move a single internal peripheral bus-mapped operand to/from internal SRAM. The same timing assumptions used in the previous example apply to this calculation. In particular, this metric also reflects the time required to activate the channel.

The eDMA design supports the following hardware service request sequence. Note that the exact timing from Cycle 7 is a function of the response times for the channel's read and write accesses. In the case of an internal peripheral bus read and internal SRAM write, the combined data phase time is 4 cycles. For an SRAM read and internal peripheral bus write, it is 5 cycles.

Table 21-1640. Hardware service request process

Cycle		Description
With internal peripheral bus read and internal SRAM write	With SRAM read and internal peripheral bus write	
1		eDMA peripheral request is asserted.
2		The eDMA peripheral request is registered locally in the eDMA module and qualified. TCD*n*_CSR[START] bit initiated requests start at this point with the registering of the user write to TCD*n* word 7.
3		Channel arbitration begins.
4		Channel arbitration completes. The transfer control descriptor local memory read is initiated.

Table continues on the next page...

Table 21-1640. Hardware service request process (continued)

Cycle		Description
With internal peripheral bus read and internal SRAM write	With SRAM read and internal peripheral bus write	
5–6		The first two parts of the activated channel's TCD is read from the local memory. The memory width to the eDMA engine is 64 bits, so the entire descriptor can be accessed in four cycles
7		The first system bus read cycle is initiated, as the third part of the channel's TCD is read from the local memory. Depending on the state of the crossbar switch, arbitration at the system bus may insert an additional cycle of delay here.
8–11	8–12	The last part of the TCD is read in. This cycle represents the first data phase for the read, and the address phase for the destination write.
12	13	This cycle represents the data phase of the last destination write.
13	14	The eDMA engine completes the execution of the inner minor loop and prepares to write back the required TCD*n* fields into the local memory. The TCD*n* word 7 is read and checked for channel linking or scatter/gather requests.
14	15	The appropriate fields in the first part of the TCD*n* are written back into the local memory.
15	16	The fields in the second part of the TCD*n* are written back into the local memory. This cycle coincides with the next channel arbitration cycle start.
16	17	The next channel to be activated performs the read of the first part of its TCD from the local memory. This is equivalent to Cycle 4 for the first channel's service request.

Assuming zero wait states on the system bus, DMA requests can be processed every 9 cycles. Assuming an average of the access times associated with internal peripheral bus-to-SRAM (4 cycles) and SRAM-to-internal peripheral bus (5 cycles), DMA requests can be processed every 11.5 cycles (4 + (4+5)/2 + 3). This is the time from Cycle 4 to Cycle x +5. The resulting peak request rate, as a function of the system frequency, is shown in the following table.

Table 21-1641. eDMA peak request rate (MReq/sec)

System frequency (MHz)	Request rate with zero wait states	Request rate with wait states
66.6	7.4	5.8
83.3	9.2	7.2
100.0	11.1	8.7
133.3	14.8	11.6
150.0	16.6	13.0

A general formula to compute the peak request rate with overlapping requests is:

PEAKreq = freq / [entry + (1 + read_ws) + (1 + write_ws) + exit]

where:

Table 21-1642. Peak request formula operands

Operand	Description
PEAKreq	Peak request rate
freq	System frequency
entry	Channel startup (4 cycles)
read_ws	Wait states seen during the system bus read data phase
write_ws	Wait states seen during the system bus write data phase
exit	Channel shutdown (3 cycles)

21.4.4.3 eDMA performance example

Consider a system with the following characteristics:

- Internal SRAM can be accessed with one wait-state when viewed from the system bus data phase
- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states viewed from the system bus data phase
- System operates at 150 MHz

For an SRAM to internal peripheral bus transfer,

PEAKreq = 150 MHz / [4 + (1 + 1) + (1 + 3) + 3] cycles = 11.5 Mreq/sec

For an internal peripheral bus to SRAM transfer,

PEAKreq = 150 MHz / [4 + (1 + 2) + (1 + 1) + 3] cycles = 12.5 Mreq/sec

Assuming an even distribution of the two transfer types, the average peak request rate would be:

PEAKreq = (11.5 Mreq/sec + 12.5 Mreq/sec) / 2 = 12.0 Mreq/sec

The minimum number of cycles to perform a single read/write, zero wait states on the system bus, from a cold start where no channel is executing and eDMA is idle are:

- 11 cycles for a software, that is, a TCD*n*_CSR[START] bit, request
- 12 cycles for a hardware, that is, an eDMA peripheral request signal, request

Two cycles account for the arbitration pipeline and one extra cycle on the hardware request resulting from the internal registering of the eDMA peripheral request signals. For the peak request rate calculations above, the arbitration and request registering is absorbed in or overlaps the previous executing channel.

Note

When channel linking or scatter/gather is enabled, a two cycle delay is imposed on the next channel selection and startup. This allows the link channel or the scatter/gather channel to be eligible and considered in the arbitration pool for next channel selection.

21.5 Initialization/application information

The following sections discuss initialization of the eDMA and programming considerations.

21.5.1 eDMA initialization

To initialize the eDMA:

1. Write to the CR if a configuration other than the default is desired.
2. Write the channel priority levels to the DCHPRI*n* registers if a configuration other than the default is desired.
3. Enable error interrupts in the EEI register if so desired.
4. Write the 32-byte TCD for each channel that may request service.
5. Enable any hardware service requests via the ERQ register.
6. Request channel service via either:
   - Software: setting the TCD*n*_CSR[START]
   - Hardware: slave device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in the following table, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, as defined by TCD*n*_SADDR, to the destination, as defined by TCD*n*_DADDR, continue until the number of bytes specified by TCD*n*_NBYTES are transferred.

When the transfer is complete, the eDMA engine's local TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, further post processing executes, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 21-1643. TCD Control and Status fields

TCD*n*_CSR field name	Description
START	Control bit to start channel explicitly when using a software initiated DMA service (Automatically cleared by hardware)
ACTIVE	Status bit indicating the channel is currently in execution
DONE	Status bit indicating major loop completion (cleared by software when using a software initiated DMA service)
D_REQ	Control bit to disable DMA request at end of major loop completion when using a hardware initiated DMA service
BWC	Control bits for throttling bandwidth control of a channel
E_SG	Control bit to enable scatter-gather feature
INT_HALF	Control bit to enable interrupt when major loop is half complete
INT_MAJ	Control bit to enable interrupt when major loop completes

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

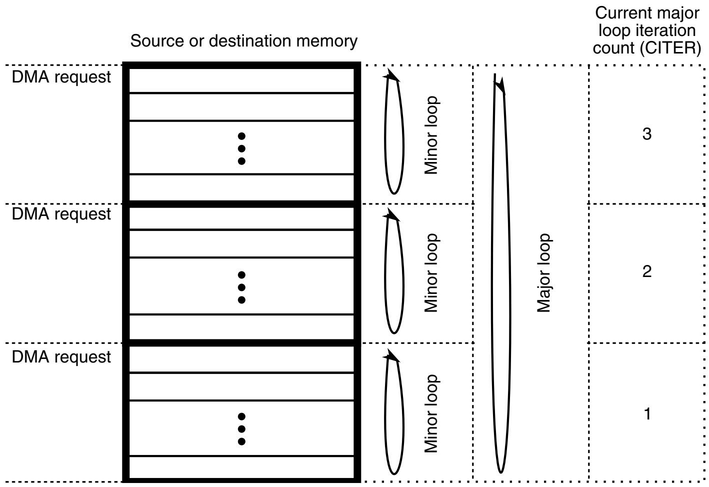


Figure 21-1637. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings interrelate.

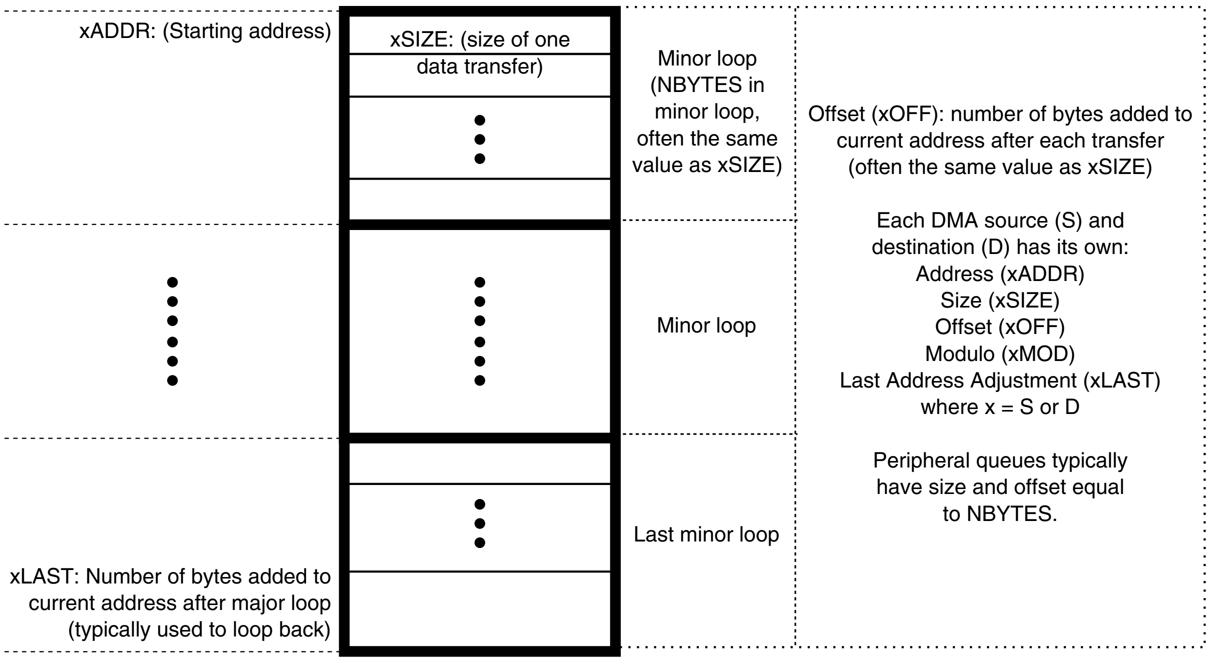


Figure 21-1638. Memory array terms

21.5.2 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data. Most programming errors are reported on a per channel basis with the exception of channel priority error (ES[CPE]).

For all error types other than group or channel priority errors, the channel number causing the error is recorded in the Error Status register (DMAx_ES). If the error source is not removed before the next activation of the problem channel, the error is detected and recorded again.

Channel priority errors are identified within a group once that group has been selected as the active group. For example:

1. The eDMA is configured for fixed group and fixed channel arbitration modes.
2. Group 1 is the highest priority and all channels are unique in that group.
3. Group 0 is the next highest priority and has two channels with the same priority level.
4. If Group 1 has any service requests, those requests will be executed.
5. After all of Group 1 requests have completed, Group 0 will be the next active group.
6. If Group 0 has a service request, then an undefined channel in Group 0 will be selected and a channel priority error will occur.
7. This repeats until the all of Group 0 requests have been removed or a higher priority Group 1 request comes in.

In this sequence, for item 2, the eDMA acknowledge lines will assert only if the selected channel is requesting service via the eDMA peripheral request signal. If interrupts are enabled for all channels, the user will get an error interrupt, but the channel number for the ERR register and the error interrupt request line may be wrong because they reflect the selected channel. A group priority error is global and any request in any group will cause a group priority error.

If priority levels are not unique, when any channel requests service, a channel priority error is reported. The highest channel/group priority with an active request is selected, but the lowest numbered channel with that priority is selected by arbitration and executed by the eDMA engine. The hardware service request handshake signals, error interrupts, and error reporting is associated with the selected channel.

21.5.3 Arbitration mode considerations

21.5.3.1 Fixed group arbitration, Fixed channel arbitration

In this mode, the channel service request from the highest priority channel in the highest priority group is selected to execute. If the eDMA is programmed so that the channels within one group use "fixed" priorities, and that group is assigned the highest "fixed" priority of all groups, that group can take all the bandwidth of the eDMA controller. That is, no other groups will be serviced if there is always at least one DMA request pending on a channel in the highest priority group when the controller arbitrates the next DMA request. The advantage of this scenario is that latency can be small for channels that need to be serviced quickly. Preemption is available in this scenario only.

21.5.3.2 Fixed group arbitration, Round-robin channel arbitration

The highest priority group with a request will be serviced. Lower priority groups will be serviced if no pending requests exist in the higher priority groups.

Within each group, channels are serviced starting with the highest channel number and rotating through to the lowest channel number without regard to the channel priority levels assigned within the group.

This scenario could cause the same bandwidth consumption problem as indicated in Fixed group arbitration, Fixed channel arbitration, but all the channels in the highest priority group will be serviced. Service latency will be short on the highest priority group, but could potentially be very much longer as the group priority decreases.

21.5.4 Performing DMA transfers (examples)

21.5.4.1 Single request

To perform a simple transfer of n bytes of data with one activation, set the major loop to one (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, the TCD*n*_CSR[DONE] bit is set and an interrupt generates if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte wide memory port located at 0x1000. The destination memory has a 32-bit port located at 0x2000. The address offsets are

programmed in increments to match the transfer size: one byte for the source and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INT_MAJ] = 1
TCDn_CSR[START] = 1 (Should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the TCD*n*_CSR[START] bit requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
    a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
    b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
    c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
    d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
    e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
    f. Write 32-bits to location 0x2008 → third iteration of the minor loop.
    g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
    h. Write 32-bits to location 0x200C → last iteration of the minor loop → major loop complete.

6. The eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 1 (TCD*n*_BITER).
7. The eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[*n*] = 1.
8. The channel retires and the eDMA goes idle or services the next channel.

21.5.4.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop transferring 16 bytes per iteration. After the channel's hardware requests are enabled in the ERQ register, the slave device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware, that is, eDMA peripheral, request for channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD*n* data from local memory to internal register file.
5. The source to destination transfers are executed as follows:
    a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
    b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
    c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
    d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
    e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
    f. Write 32-bits to location 0x2008 → third iteration of the minor loop.

    g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
    h. Write 32-bits to location 0x200C → last iteration of the minor loop.
6. eDMA engine writes: TCD*n*_SADDR = 0x1010, TCD*n*_DADDR = 0x2010, TCD*n*_CITER = 1.
7. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0.
8. The channel retires → one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware, that is, eDMA peripheral, requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
12. eDMA engine reads: channel TCD data from local memory to internal register file.
13. The source to destination transfers are executed as follows:
    a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.
    b. Write 32-bits to location 0x2010 → first iteration of the minor loop.
    c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.
    d. Write 32-bits to location 0x2014 → second iteration of the minor loop.
    e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.
    f. Write 32-bits to location 0x2018 → third iteration of the minor loop.
    g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.
    h. Write 32-bits to location 0x201C → last iteration of the minor loop → major loop complete.
14. eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 2 (TCD*n*_BITER).
15. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[n] = 1.

16. The channel retires → major loop complete. The eDMA goes idle or services the next channel.

21.5.4.3 Using the modulo feature

The modulo feature of the eDMA provides the ability to implement a circular data queue in which the size of the queue is a power of 2. MOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of 0 for this field disables the modulo feature.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value while the 28 upper address bits (0x1234567*x*) retain their original value. In this example the source address is set to 0x12345670, the offset is set to 4 bytes and the MOD field is set to 4, allowing for a 2^4 byte (16-byte) size queue.

Table 21-1644. Modulo example

Transfer Number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670
6	0x12345674

21.5.5 Monitoring transfer descriptor status

21.5.5.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software initiated service requests. The first is to read the TCD*n*_CITER field and test for a change. Another method may be extracted from the sequence shown below. The second method is to test the TCD*n*_CSR[START] bit and the TCD*n*_CSR[ACTIVE] bit. The minor-loop-complete condition is indicated by both bits reading zero after the TCD*n*_CSR[START] was set. Polling the TCD*n*_CSR[ACTIVE] bit may be inconclusive, because the active status may be missed if the channel execution is short in duration.

The TCD status bits execute the following sequence for a software activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	1	0	0	Channel service request via software
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

The best method to test for minor-loop completion when using hardware, that is, peripheral, initiated service requests is to read the TCD*n*_CITER field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status bits execute the following sequence for a hardware-activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	0	0	0	Channel service request via hardware (peripheral request asserted)
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

For both activation types, the major-loop-complete status is explicitly indicated via the TCD*n*_CSR[DONE] bit.

The TCD*n*_CSR[START] bit is cleared automatically when the channel begins execution regardless of how the channel activates.

21.5.5.2 Reading the transfer descriptors of active channels

The eDMA reads back the true TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_NBYTES values if read while a channel executes. The true values of the SADDR, DADDR, and NBYTES are the values the eDMA engine currently uses in its internal register file and not the values in the TCD local memory for that channel. The addresses, SADDR and DADDR, and NBYTES, which decrement to zero as the transfer progresses, can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

21.5.5.3 Checking channel preemption status

Preemption is available only when fixed arbitration is selected for both group and channel arbitration modes. A preemptive situation is one in which a preempt-enabled channel runs and a higher priority request becomes active. When the eDMA engine is not operating in fixed group, fixed channel arbitration mode, the determination of the actively running relative priority outstanding requests become undefined. Channel and/or group priorities are treated as equal, that is, constantly rotating, when Round-Robin Arbitration mode is selected.

The TCD*n*_CSR[ACTIVE] bit for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended while the preempting channel executes one major loop iteration. If two TCD*n*_CSR[ACTIVE] bits are set simultaneously in the global TCD map, a higher priority channel is actively preempting a lower priority channel.

21.5.6 Channel Linking

Channel linking (or chaining) is a mechanism where one channel sets the TCD*n*_CSR[START] bit of another channel (or itself), therefore initiating a service request for that channel. When properly enabled, the EDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCD*n*_CITER[E_LINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, the initial fields of:

```
TCDn_CITER[E_LINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJOR_E_LINK] = 1
TCDn_CSR[MAJOR_LINKCH] = 0x7
```

executes as:

1. Minor loop done → set TCD12_CSR[START] bit
2. Minor loop done → set TCD12_CSR[START] bit
3. Minor loop done → set TCD12_CSR[START] bit
4. Minor loop done, major loop done→ set TCD7_CSR[START] bit

When minor loop linking is enabled (TCD*n*_CITER[E_LINK] = 1), the TCD*n*_CITER[CITER] field uses a nine bit vector to form the current iteration count. When minor loop linking is disabled (TCD*n*_CITER[E_LINK] = 0), the TCD*n*_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCD*n*_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.

Note

The TCD*n*_CITER[E_LINK] bit and the TCD*n*_BITER[E_LINK] bit must equal or a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop, half-way done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, i.e, use another channel's TCD, at the end of a loop.

Table 21-1645. Channel Linking Parameters

Desired Link Behavior	TCD Control Field Name	Description
Link at end of Minor Loop	CITER[E_LINK]	Enable channel-to-channel linking on minor loop completion (current iteration)
	CITER[LINKCH]	Link channel number when linking at end of minor loop (current iteration)
Link at end of Major Loop	CSR[MAJOR_E_LINK]	Enable channel-to-channel linking on major loop completion
	CSR[MAJOR_LINKCH]	Link channel number when linking at end of major loop

21.5.7 Dynamic programming

21.5.7.1 Dynamically changing the channel priority

The following two options are recommended for dynamically changing channel priority levels:

1. Switch to Round-Robin Channel Arbitration mode, change the channel priorities, then switch back to Fixed Arbitration mode,
2. Disable all the channels, change the channel priorities, then enable the appropriate channels.

21.5.7.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCD.major.e_link bit during channel execution. This bit is read from the TCD local memory at the end of channel execution, thus allowing the user to enable the feature during channel execution.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic channel link by enabling the TCD.major.e_link bit at the same time the eDMA engine is retiring the channel. The TCD.major.e_link would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

The following coherency model is recommended when executing a dynamic channel link request.

1. Write 1 to the TCD.major.e_link bit.
2. Read back the TCD.major.e_link bit.
3. Test the TCD.major.e_link request status:
    - If TCD.major.e_link = 1, the dynamic link attempt was successful.
    - If TCD.major.e_link = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCD.major.e_link bit to zero on any writes to a channel's TCD.word7 after that channel's TCD.done bit is set, indicating the major loop is complete.

NOTE

The user must clear the TCD.done bit before writing the TCD.major.e_link bit. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

21.5.7.3 Dynamic scatter/gather

Scatter/gather is the process of automatically loading a new TCD into a channel. It allows a DMA channel to use multiple TCDs; this enables a DMA channel to scatter the DMA data to multiple destinations or gather it from multiple sources.When scatter/gather is enabled and the channel has finished its major loop, a new TCD is fetched from system memory and loaded into that channel's descriptor location in eDMA programmer's model, thus replacing the current descriptor.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic scatter/gather operation by enabling the TCD.e_sg bit at the same time the eDMA engine

is retiring the channel. The TCD.e_sg would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods for this coherency model are shown in the following subsections. Method 1 has the advantage of reading the major.linkch field and the e_sg bit with a single read. For both dynamic channel linking and scatter/gather requests, the TCD local memory controller forces the TCD.major.e_link and TCD.e_sg bits to zero on any writes to a channel's TCD.word7 if that channel's TCD.done bit is set indicating the major loop is complete.

NOTE

The user must clear the TCD.done bit before writing the TCD.major.e_link or TCD.e_sg bits. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

21.5.7.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When the TCD.major.e_link bit is zero, the TCD.major.linkch field is not used by the eDMA. In this case, the TCD.major.linkch bits may be used for other purposes. This method uses the TCD.major.linkch field as a TCD indentification (ID).

1. When the descriptors are built, write a unique TCD ID in the TCD.major.linkch field for each TCD associated with a channel using dynamic scatter/gather.
2. Write 1b to the TCD.d_req bit.

   Should a dynamic scatter/gather attempt fail, setting the TCD.d_req bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offest value.

3. Write the TCD.dlast_sga field with the scatter/gather address.
4. Write 1b to the TCD.e_sg bit.
5. Read back the 16 bit TCD control/status field.
6. Test the TCD.e_sg request status and TCD.major.linkch value:

   If e_sg = 1b, the dynamic link attempt was successful.

If e_sg = 0b and the major.linkch (ID) did not change, the attempted dynamic link did not succeed (the channel was already retiring).

If e_sg = 0b and the major.linkch (ID) changed, the dynamic link attempt was successful (the new TCD's e_sg value cleared the e_sg bit).

21.5.7.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCD.dlast_sga field as a TCD indentification (ID).

1. Write 1b to the TCD.d_req bit.

   Should a dynamic scatter/gather attempt fail, setting the d_req bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offest value.

2. Write theTCD.dlast_sga field with the scatter/gather address.
3. Write 1b to the TCD.e_sg bit.
4. Read back the TCD.e_sg bit.
5. Test the TCD.e_sg request status:

   If e_sg = 1b, the dynamic link attempt was successful.

   If e_sg = 0b, read the 32 bit TCD dlast_sga field.

   If e_sg = 0b and the dlast_sga did not change, the attempted dynamic link did not succeed (the channel was already retiring).

   If e_sg = 0b and the dlast_sga changed, the dynamic link attempt was successful (the new TCD's e_sg value cleared the e_sg bit).

Chapter 22
Direct Memory Access Multiplexer (DMAMUX)

22.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

22.1.1 Overview

The direct memory access multiplexer (DMAMUX) routes DMA sources, called slots, to any of the 16 DMA channels. This process is illustrated in the following figure.

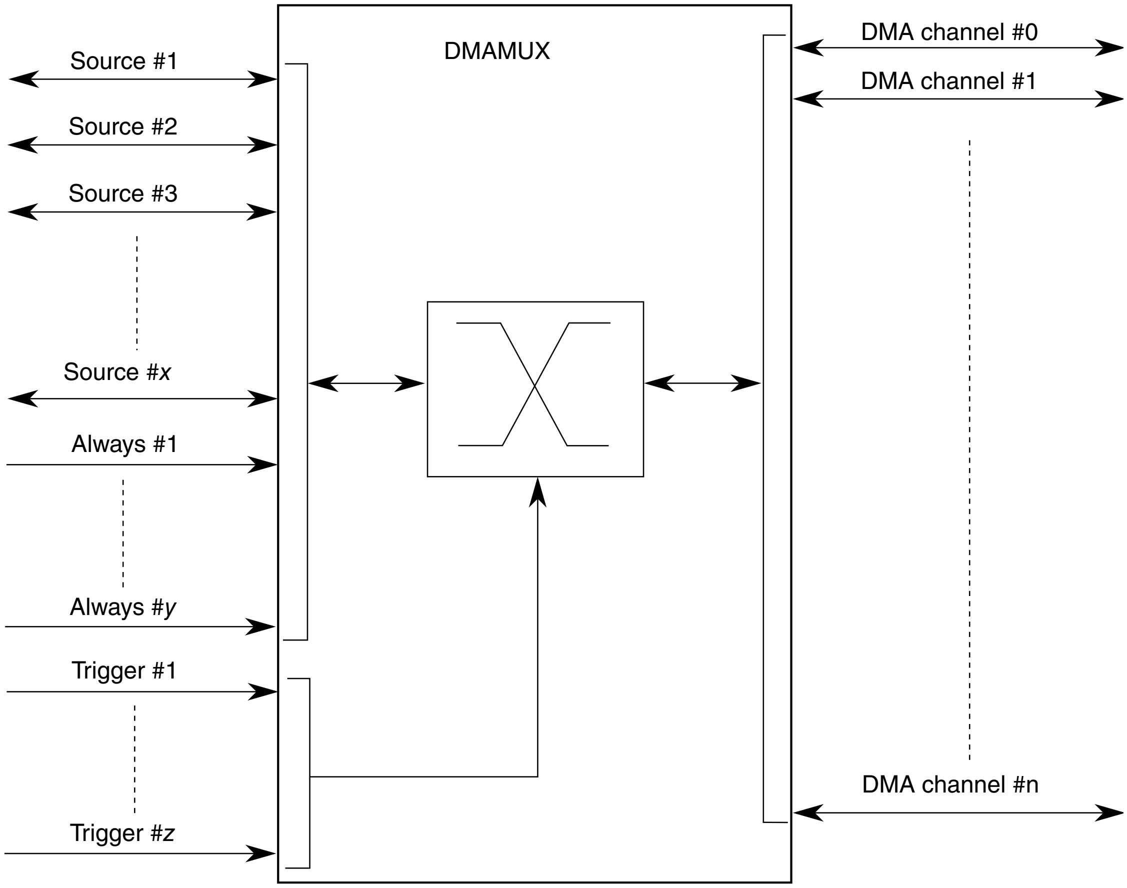


Figure 22-1. DMAMUX block diagram

22.1.2 Features

The DMAMUX module provides these features:

- Up to 53 peripheral slots and up to 10 always-on slots can be routed to 16 channels.
- 16 independently selectable DMA channel routers.
    - The first four channels additionally provide a trigger functionality.
- Each channel router can be assigned to one of the possible peripheral DMA slots or to one of the always-on slots.

22.1.3 Modes of operation

The following operating modes are available:

- Disabled mode

In this mode, the DMA channel is disabled. Because disabling and enabling of DMA channels is done primarily via the DMA configuration registers, this mode is used mainly as the reset state for a DMA channel in the DMA channel MUX. It may also be used to temporarily suspend a DMA channel while reconfiguration of the system takes place, for example, changing the period of a DMA trigger.

- Normal mode

  In this mode, a DMA source is routed directly to the specified DMA channel. The operation of the DMAMUX in this mode is completely transparent to the system.

- Periodic Trigger mode

  In this mode, a DMA source may only request a DMA transfer, such as when a transmit buffer becomes empty or a receive buffer becomes full, periodically. Configuration of the period is done in the registers of the periodic interrupt timer (PIT). This mode is available only for channels 0–3.

22.2 External signal description

The DMAMUX has no external pins.

22.3 Memory map/register definition

This section provides a detailed description of all memory-mapped registers in the DMAMUX.

DMAMUX memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_4000	Channel Configuration register (DMAMUX0_CHCFG0)	8	R/W	00h	22.3.1/1067
4002_4001	Channel Configuration register (DMAMUX0_CHCFG1)	8	R/W	00h	22.3.1/1067
4002_4002	Channel Configuration register (DMAMUX0_CHCFG2)	8	R/W	00h	22.3.1/1067
4002_4003	Channel Configuration register (DMAMUX0_CHCFG3)	8	R/W	00h	22.3.1/1067
4002_4004	Channel Configuration register (DMAMUX0_CHCFG4)	8	R/W	00h	22.3.1/1067
4002_4005	Channel Configuration register (DMAMUX0_CHCFG5)	8	R/W	00h	22.3.1/1067
4002_4006	Channel Configuration register (DMAMUX0_CHCFG6)	8	R/W	00h	22.3.1/1067
4002_4007	Channel Configuration register (DMAMUX0_CHCFG7)	8	R/W	00h	22.3.1/1067
4002_4008	Channel Configuration register (DMAMUX0_CHCFG8)	8	R/W	00h	22.3.1/1067

Table continues on the next page...

DMAMUX memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_4009	Channel Configuration register (DMAMUX0_CHCFG9)	8	R/W	00h	22.3.1/1067
4002_400A	Channel Configuration register (DMAMUX0_CHCFG10)	8	R/W	00h	22.3.1/1067
4002_400B	Channel Configuration register (DMAMUX0_CHCFG11)	8	R/W	00h	22.3.1/1067
4002_400C	Channel Configuration register (DMAMUX0_CHCFG12)	8	R/W	00h	22.3.1/1067
4002_400D	Channel Configuration register (DMAMUX0_CHCFG13)	8	R/W	00h	22.3.1/1067
4002_400E	Channel Configuration register (DMAMUX0_CHCFG14)	8	R/W	00h	22.3.1/1067
4002_400F	Channel Configuration register (DMAMUX0_CHCFG15)	8	R/W	00h	22.3.1/1067
4002_5000	Channel Configuration register (DMAMUX1_CHCFG0)	8	R/W	00h	22.3.1/1067
4002_5001	Channel Configuration register (DMAMUX1_CHCFG1)	8	R/W	00h	22.3.1/1067
4002_5002	Channel Configuration register (DMAMUX1_CHCFG2)	8	R/W	00h	22.3.1/1067
4002_5003	Channel Configuration register (DMAMUX1_CHCFG3)	8	R/W	00h	22.3.1/1067
4002_5004	Channel Configuration register (DMAMUX1_CHCFG4)	8	R/W	00h	22.3.1/1067
4002_5005	Channel Configuration register (DMAMUX1_CHCFG5)	8	R/W	00h	22.3.1/1067
4002_5006	Channel Configuration register (DMAMUX1_CHCFG6)	8	R/W	00h	22.3.1/1067
4002_5007	Channel Configuration register (DMAMUX1_CHCFG7)	8	R/W	00h	22.3.1/1067
4002_5008	Channel Configuration register (DMAMUX1_CHCFG8)	8	R/W	00h	22.3.1/1067
4002_5009	Channel Configuration register (DMAMUX1_CHCFG9)	8	R/W	00h	22.3.1/1067
4002_500A	Channel Configuration register (DMAMUX1_CHCFG10)	8	R/W	00h	22.3.1/1067
4002_500B	Channel Configuration register (DMAMUX1_CHCFG11)	8	R/W	00h	22.3.1/1067
4002_500C	Channel Configuration register (DMAMUX1_CHCFG12)	8	R/W	00h	22.3.1/1067
4002_500D	Channel Configuration register (DMAMUX1_CHCFG13)	8	R/W	00h	22.3.1/1067
4002_500E	Channel Configuration register (DMAMUX1_CHCFG14)	8	R/W	00h	22.3.1/1067
4002_500F	Channel Configuration register (DMAMUX1_CHCFG15)	8	R/W	00h	22.3.1/1067
400A_1000	Channel Configuration register (DMAMUX2_CHCFG0)	8	R/W	00h	22.3.1/1067
400A_1001	Channel Configuration register (DMAMUX2_CHCFG1)	8	R/W	00h	22.3.1/1067
400A_1002	Channel Configuration register (DMAMUX2_CHCFG2)	8	R/W	00h	22.3.1/1067
400A_1003	Channel Configuration register (DMAMUX2_CHCFG3)	8	R/W	00h	22.3.1/1067
400A_1004	Channel Configuration register (DMAMUX2_CHCFG4)	8	R/W	00h	22.3.1/1067
400A_1005	Channel Configuration register (DMAMUX2_CHCFG5)	8	R/W	00h	22.3.1/1067
400A_1006	Channel Configuration register (DMAMUX2_CHCFG6)	8	R/W	00h	22.3.1/1067
400A_1007	Channel Configuration register (DMAMUX2_CHCFG7)	8	R/W	00h	22.3.1/1067
400A_1008	Channel Configuration register (DMAMUX2_CHCFG8)	8	R/W	00h	22.3.1/1067
400A_1009	Channel Configuration register (DMAMUX2_CHCFG9)	8	R/W	00h	22.3.1/1067
400A_100A	Channel Configuration register (DMAMUX2_CHCFG10)	8	R/W	00h	22.3.1/1067
400A_100B	Channel Configuration register (DMAMUX2_CHCFG11)	8	R/W	00h	22.3.1/1067
400A_100C	Channel Configuration register (DMAMUX2_CHCFG12)	8	R/W	00h	22.3.1/1067
400A_100D	Channel Configuration register (DMAMUX2_CHCFG13)	8	R/W	00h	22.3.1/1067
400A_100E	Channel Configuration register (DMAMUX2_CHCFG14)	8	R/W	00h	22.3.1/1067

Table continues on the next page...

DMAMUX memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_100F	Channel Configuration register (DMAMUX2_CHCFG15)	8	R/W	00h	22.3.1/1067
400A_2000	Channel Configuration register (DMAMUX3_CHCFG0)	8	R/W	00h	22.3.1/1067
400A_2001	Channel Configuration register (DMAMUX3_CHCFG1)	8	R/W	00h	22.3.1/1067
400A_2002	Channel Configuration register (DMAMUX3_CHCFG2)	8	R/W	00h	22.3.1/1067
400A_2003	Channel Configuration register (DMAMUX3_CHCFG3)	8	R/W	00h	22.3.1/1067
400A_2004	Channel Configuration register (DMAMUX3_CHCFG4)	8	R/W	00h	22.3.1/1067
400A_2005	Channel Configuration register (DMAMUX3_CHCFG5)	8	R/W	00h	22.3.1/1067
400A_2006	Channel Configuration register (DMAMUX3_CHCFG6)	8	R/W	00h	22.3.1/1067
400A_2007	Channel Configuration register (DMAMUX3_CHCFG7)	8	R/W	00h	22.3.1/1067
400A_2008	Channel Configuration register (DMAMUX3_CHCFG8)	8	R/W	00h	22.3.1/1067
400A_2009	Channel Configuration register (DMAMUX3_CHCFG9)	8	R/W	00h	22.3.1/1067
400A_200A	Channel Configuration register (DMAMUX3_CHCFG10)	8	R/W	00h	22.3.1/1067
400A_200B	Channel Configuration register (DMAMUX3_CHCFG11)	8	R/W	00h	22.3.1/1067
400A_200C	Channel Configuration register (DMAMUX3_CHCFG12)	8	R/W	00h	22.3.1/1067
400A_200D	Channel Configuration register (DMAMUX3_CHCFG13)	8	R/W	00h	22.3.1/1067
400A_200E	Channel Configuration register (DMAMUX3_CHCFG14)	8	R/W	00h	22.3.1/1067
400A_200F	Channel Configuration register (DMAMUX3_CHCFG15)	8	R/W	00h	22.3.1/1067

22.3.1 Channel Configuration register (DMAMUXx_CHCFGn)

Each of the DMA channels can be independently enabled/disabled and associated with one of the DMA slots (peripheral slots or always-on slots) in the system.

NOTE

Setting multiple CHCFG registers with the same source value will result in unpredictable behavior.

Before changing the trigger or source settings, a DMA channel must be disabled via the CHCFGn[ENBL] bit.

Address: Base address + 0h offset + (1d × i), where i=0d to 15d

Bit	7	6	5	4	3	2	1	0
Read Write	ENBL	TRIG	SOURCE					
Reset	0	0	0	0	0	0	0	0

DMAMUXx_CHCFGn field descriptions

Field	Description
7 ENBL	DMA Channel Enable Enables the DMA channel. 0 DMA channel is disabled. This mode is primarily used during configuration of the DMAMux. The DMA has separate channel enables/disables, which should be used to disable or reconfigure a DMA channel. 1 DMA channel is enabled
6 TRIG	DMA Channel Trigger Enable Enables the periodic trigger capability for the triggered DMA channel. 0 Triggering is disabled. If triggering is disabled and the ENBL bit is set, the DMA Channel will simply route the specified source to the DMA channel. (Normal mode) 1 Triggering is enabled. If triggering is enabled and the ENBL bit is set, the DMAMUX is in Periodic Trigger mode.
5–0 SOURCE	DMA Channel Source (Slot) Specifies which DMA source, if any, is routed to a particular DMA channel. See your device's chip configuration details for information about the peripherals and their slot numbers.

22.4 Functional description

The primary purpose of the DMAMUX is to provide flexibility in the system's use of the available DMA channels. As such, configuration of the DMAMUX is intended to be a static procedure done during execution of the system boot code. However, if the procedure outlined in Enabling and configuring sources is followed, the configuration of the DMAMUX may be changed during the normal operation of the system.

Functionally, the DMAMUX channels may be divided into two classes:

- Channels that implement the normal routing functionality plus periodic triggering capability
- Channels that implement only the normal routing functionality

22.4.1 DMA channels with periodic triggering capability

Besides the normal routing functionality, the first 4 channels of the DMAMUX provide a special periodic triggering capability that can be used to provide an automatic mechanism to transmit bytes, frames, or packets at fixed intervals without the need for processor intervention. The trigger is generated by the periodic interrupt timer (PIT); as such, the configuration of the periodic triggering interval is done via configuration registers in the PIT. See the section on periodic interrupt timer for more information on this topic.

Note

Because of the dynamic nature of the system (due to DMA channel priorities, bus arbitration, interrupt service routine lengths, etc.), the number of clock cycles between a trigger and the actual DMA transfer cannot be guaranteed.

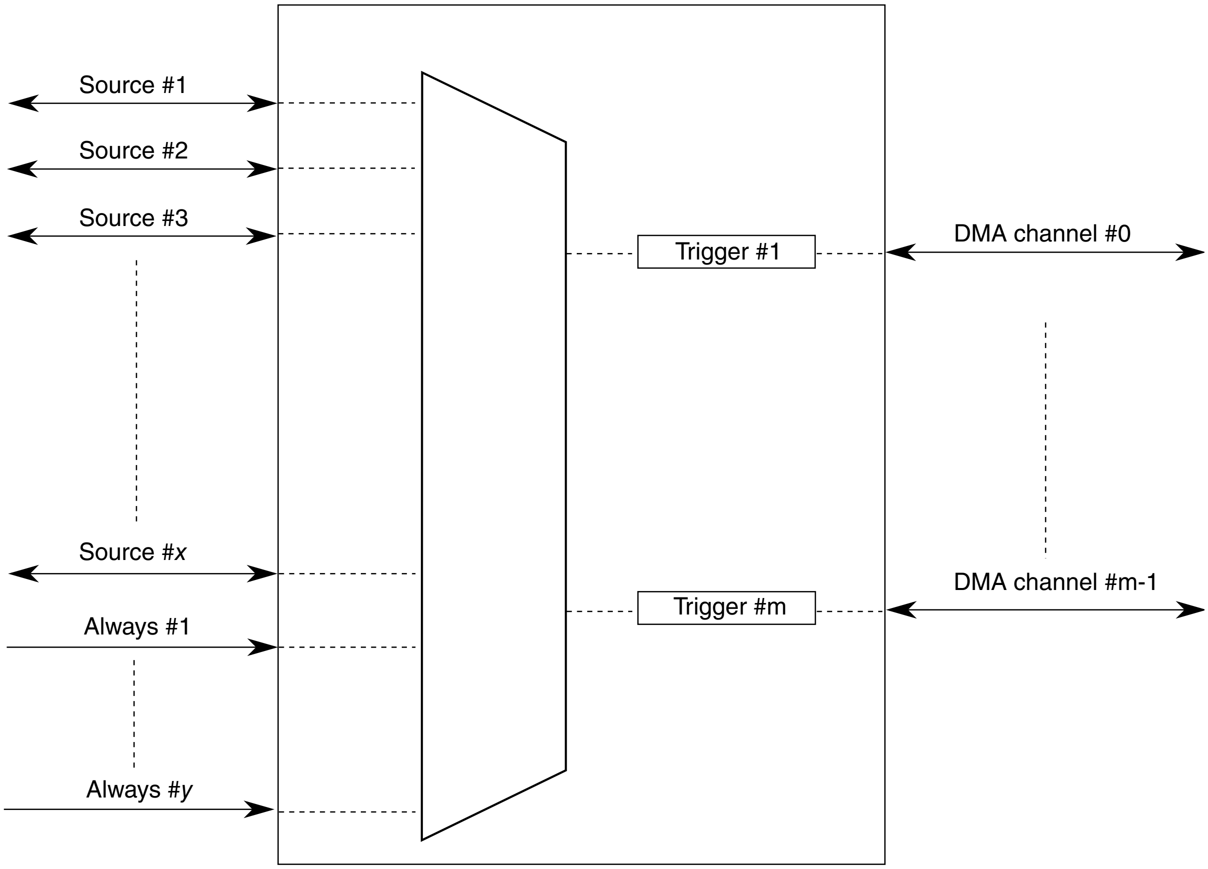


Figure 22-87. DMAMUX triggered channels

The DMA channel triggering capability allows the system to schedule regular DMA transfers, usually on the transmit side of certain peripherals, without the intervention of the processor. This trigger works by gating the request from the peripheral to the DMA until a trigger event has been seen. This is illustrated in the following figure.

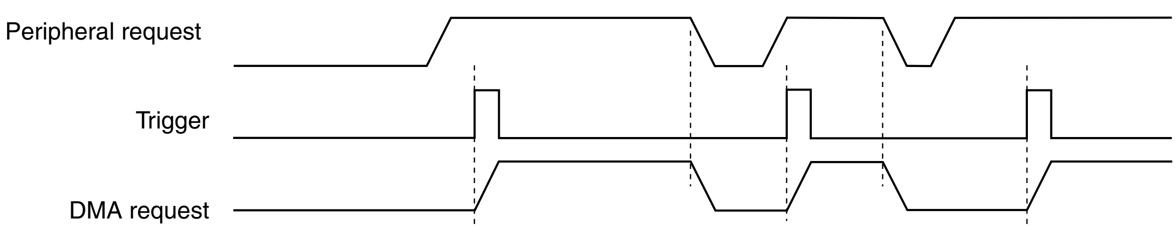


Figure 22-88. DMAMUX channel triggering: normal operation

After the DMA request has been serviced, the peripheral will negate its request, effectively resetting the gating mechanism until the peripheral reasserts its request and the next trigger event is seen. This means that if a trigger is seen, but the peripheral is not requesting a transfer, then that trigger will be ignored. This situation is illustrated in the following figure.

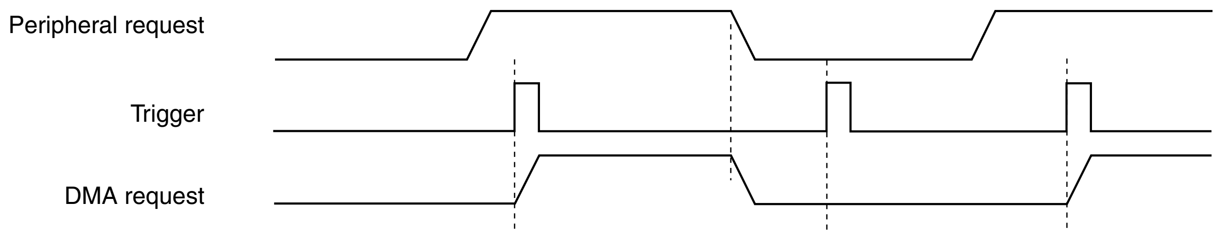


Figure 22-89. DMAMUX channel triggering: ignored trigger

This triggering capability may be used with any peripheral that supports DMA transfers, and is most useful for two types of situations:

- Periodically polling external devices on a particular bus

  As an example, the transmit side of an SPI is assigned to a DMA channel with a trigger, as described above. After it has been set up, the SPI will request DMA transfers, presumably from memory, as long as its transmit buffer is empty. By using a trigger on this channel, the SPI transfers can be automatically performed every 5 μs (as an example). On the receive side of the SPI, the SPI and DMA can be configured to transfer receive data into memory, effectively implementing a method to periodically read data from external devices and transfer the results into memory without processor intervention.

- Using the GPIO ports to drive or sample waveforms

  By configuring the DMA to transfer data to one or more GPIO ports, it is possible to create complex waveforms using tabular data stored in on-chip memory. Conversely, using the DMA to periodically transfer data from one or more GPIO ports, it is possible to sample complex waveforms and store the results in tabular form in on-chip memory.

A more detailed description of the capability of each trigger, including resolution, range of values, and so on, may be found in the periodic interrupt timer section.

22.4.2 DMA channels with no triggering capability

The other channels of the DMAMUX provide the normal routing functionality as described in Modes of operation.

22.4.3 Always-enabled DMA sources

In addition to the peripherals that can be used as DMA sources, there are 10 additional DMA sources that are always enabled. Unlike the peripheral DMA sources, where the peripheral controls the flow of data during DMA transfers, the sources that are always enabled provide no such "throttling" of the data transfers. These sources are most useful in the following cases:

- Performing DMA transfers to/from GPIO—Moving data from/to one or more GPIO pins, either unthrottled (that is, as fast as possible), or periodically (using the DMA triggering capability).
- Performing DMA transfers from memory to memory—Moving data from memory to memory, typically as fast as possible, sometimes with software activation.
- Performing DMA transfers from memory to the external bus, or vice-versa—Similar to memory to memory transfers, this is typically done as quickly as possible.
- Any DMA transfer that requires software activation—Any DMA transfer that should be explicitly started by software.

In cases where software should initiate the start of a DMA transfer, an always-enabled DMA source can be used to provide maximum flexibility. When activating a DMA channel via software, subsequent executions of the minor loop require that a new start event be sent. This can either be a new software activation, or a transfer request from the DMA channel MUX. The options for doing this are:

- Transfer all data in a single minor loop.

  By configuring the DMA to transfer all of the data in a single minor loop (that is, major loop counter = 1), no reactivation of the channel is necessary. The disadvantage to this option is the reduced granularity in determining the load that the DMA transfer will impose on the system. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use explicit software reactivation.

  In this option, the DMA is configured to transfer the data using both minor and major loops, but the processor is required to reactivate the channel by writing to the DMA registers *after every minor loop*. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use an always-enabled DMA source.

In this option, the DMA is configured to transfer the data using both minor and major loops, and the DMA channel MUX does the channel reactivation. For this option, the DMA channel should be enabled and pointing to an "always enabled" source. Note that the reactivation of the channel can be continuous (DMA triggering is disabled) or can use the DMA triggering capability. In this manner, it is possible to execute periodic transfers of packets of data from one source to another, without processor intervention.

22.5 Initialization/application information

This section provides instructions for initializing the DMA channel MUX.

22.5.1 Reset

The reset state of each individual bit is shown in Memory map/register definition. In summary, after reset, all channels are disabled and must be explicitly enabled before use.

22.5.2 Enabling and configuring sources

To enable a source with periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Configure the corresponding timer.
5. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

NOTE

The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with periodic triggering capability:

1. Write 0x00 to CHCFG1 (base address + 0x01).

2. Configure channel 1 in the DMA, including enabling the channel.
3. Configure a timer for the desired trigger interval.
4. Write 0xC5 to CHCFG1 (base address + 0x01).

The following code example illustrates steps 1 and 4 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0xFC084000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCONFIG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCONFIG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCONFIG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCONFIG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCONFIG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCONFIG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCONFIG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCONFIG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCONFIG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCONFIG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCONFIG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCONFIG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCONFIG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCONFIG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCONFIG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCONFIG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);

In File main.c:
#include "registers.h"
:
:
*CHCONFIG1 = 0x00;
*CHCONFIG1 = 0xC5;
```

To enable a source without periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that CHCFG[ENBL] is set while CHCFG[TRIG] is cleared.

NOTE

The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with no periodic triggering capability:

1. Write 0x00 to CHCFG1 (base address + 0x01).
2. Configure channel 1 in the DMA, including enabling the channel.
3. Write 0x85 to CHCFG1 (base address + 0x01).

The following code example illustrates steps 1 and 3 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0xFC084000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCONFIG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCONFIG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCONFIG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCONFIG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCONFIG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCONFIG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCONFIG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCONFIG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCONFIG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCONFIG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCONFIG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCONFIG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCONFIG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCONFIG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCONFIG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCONFIG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);

In File main.c:
#include "registers.h"
:
:
*CHCONFIG1 = 0x00;
*CHCONFIG1 = 0x85;
```

To disable a source:

A particular DMA source may be disabled by not writing the corresponding source value into any of the CHCFG registers. Additionally, some module-specific configuration may be necessary. See the appropriate section for more details.

To switch the source of a DMA channel:

1. Disable the DMA channel in the DMA and reconfigure the channel for the new source.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] bits of the DMA channel.
3. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

To switch DMA channel 8 from source #5 transmit to source #7 transmit:

1. In the DMA configuration registers, disable DMA channel 8 and reconfigure it to handle the transfers to peripheral slot 7. This example assumes channel 8 doesn't have triggering capability.
2. Write 0x00 to CHCFG8 (base address + 0x08).
3. Write 0x87 to CHCFG8 (base address + 0x08). (In this example, setting CHCFG[TRIG] would have no effect due to the assumption that channel 8 does not support the periodic triggering functionality).

The following code example illustrates steps 2 and 3 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0xFC084000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCONFIG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCONFIG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCONFIG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCONFIG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCONFIG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCONFIG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCONFIG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCONFIG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCONFIG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCONFIG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCONFIG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCONFIG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCONFIG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCONFIG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCONFIG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCONFIG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);

In File main.c:
#include "registers.h"
:
:
*CHCONFIG8 = 0x00;
*CHCONFIG8 = 0x87;
```

Chapter 23
Peripheral Bridge (AIPS-Lite)

23.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The peripheral bridge converts the crossbar switch interface to an interface that can access most of the slave peripherals on this chip.

The peripheral bridge occupies 64 MB of the address space, which is divided into peripheral slots of . (It might be possible that all the peripheral slots are not used. See Memory map for details on slot assignments.) The bridge includes separate clock enable inputs for each of the slots to accommodate slower peripherals.

23.1.1 Features

Key features of the peripheral bridge are:

- Supports peripheral slots with 8-, 16-, and 32-bit datapath width
- Supports a pair of 32-bit transactions for selected 64-bit memory accesses

23.1.2 General operation

The slave devices connected to the peripheral bridge are modules which contain a programming model of control and status registers. The system masters read and write these registers through the peripheral bridge. The peripheral bridge performs a bus protocol conversion of the master transactions and generates the following as inputs to the peripherals:

- Module enables
- Module addresses
- Transfer attributes
- Byte enables
- Write data

The peripheral bridge selects and captures read data from the peripheral interface and returns it to the crossbar switch.

Each peripheral is allocated one or more block(s) of the memory map. Two global external module enables are available for the remaining address space to allow for customization and expansion of addressed peripheral devices.

The AIPS-Lite module uses the data width of accessed peripheral to perform proper data byte lane routing; no bus decomposition (bus sizing) is performed.

23.2 Functional description

The peripheral bridge functions as a bus protocol translator between the crossbar switch and the slave peripheral bus. Support is provided for generating a pair of 32-bit slave accesses when performing certain 64-bit peripheral accesses.

The peripheral bridge manages all transactions destined for the attached slave devices and generates select signals for modules on the peripheral bus by decoding accesses within the attached address space.

23.2.1 Access support

Aligned and misaligned 32-bit, 16-bit, and byte accesses are supported for 32-bit peripherals. Misaligned accesses are supported to allow memory to be placed on the slave peripheral bus. Peripheral registers must not be misaligned, although no explicit checking is performed by the peripheral bridge. The peripheral bridge performs two slave peripheral bus transfers for allowed 64-bit transactions, to 32-bit sized peripheral slots only.All other accesses are performed with a single transfer.

All accesses to the peripheral slots must be sized less than or equal to the designated peripheral slot size. If an access is attempted that is larger than the targeted port, an error response is generated.

Chapter 24
IPS_Semaphores

24.1 Introduction

The module provides the hardware support needed in multi-core systems for implementing semaphores and provide a simple mechanism to achieve "lock/unlock" operations via a single write access. This approach eliminates architecture-specific implementations like atomic (indivisible) read-modify-write instructions or reservation mechanisms. The result is an architecture-neutral solution that provides hardware-enforced gates as well as other useful system functions related to the gating mechanisms.

24.1.1 Multi-Core Programming 101: Software Gates

Multi-processor systems require a function that can be used to safely and easily provide a locking mechanism which is then used by system software to control access to shared data structures, shared hardware resources, etc. These "gating" mechanisms are used by the software to serialize (and synchronize) writes to shared data and/or resources to prevent race conditions and preserve memory coherency between different processes and processors.

Consider the following description of a typical use-case. Processor X enters a section of code where shared data values are to be updated. It must first acquire a semaphore; this can be considered "locking (or closing) a software gate." Once the gate has been locked, a properly-architected software system does not allow other processes (or processors) to execute the same code segment or modify the shared data structure protected by the gate, that is, other processes/processors are locked out. Many software implementations include a *spin-wait loop* within the lock function until the locking of the gate is accomplished. Once the lock has been obtained, Processor X continues execution and updates the data values protected by the particular lock. Once the updates are complete, Processor X unlocks (or opens) the software gate, allowing other processes/processors access to the updated data values.

There are three important rules that must be followed for a correctly-implemented system solution:

1. All writes to shared data values or shared hardware resources must be protected by a "gate" variable.
2. Once a processor locks a gate, accesses to the shared data or resources by other processes/processors must be blocked. This is enforced by software conventions.
3. The processor that locks a particular gate is the only processor that can open (unlock) that gate.

Information in the hardware gate identifying the locking processor can be extremely useful for system-level debugging.

The Hennessy/Patterson text on computer architecture offers the following description of software gating.

"One of the major requirements of a shared-memory architecture multiprocessor is being able to coordinate processes that are working on a common task. Typically, a programmer will use *lock variables* to synchronize the processes.

The difficulty for the architect of a multiprocessor is to provide a mechanism to decide which processor gets the lock and to provide the operation that locks a variable. Arbitration is easy for shared-bus multiprocessors, since the bus is the only path to memory. The processor that gets the bus locks out all the other processors from memory. If the CPU and bus provide an atomic swap operation, programmers can create locks with the proper semantics. The adjective *atomic* is key, for it means that a processor can both read a location **and** set it to the locked value in the same bus operation, preventing any other processor from reading or writing memory." [Hennessy/Patterson, *Computer Architecture: A Quantitative Approach*, ppg. 471-472]

The classic text continues with a description of the steps required to lock/unlock a variable using an *atomic swap instruction*.

"Assume that 0 means unlocked and 1 means locked. A processor first reads the lock variable to test its state. A processor keeps reading and testing until the value indicates that the lock is unlocked. The processor then races against all other processes that were similarly "spin waiting" to see who can lock the variable first. All processes use a swap instruction that reads the old value and stores a 1 into the lock variable. The single winner will see the 0, and the losers will see a 1 that was placed there by the winner. (The losers will continue to set the variable to the locked value, but that doesn't matter.) The winning processor executes the code after the lock and then stores a 0 into the lock when it exits,

starting the race all over again. Testing the old value and then setting to a new value is why the atomic swap instruction is called *test and set* in some instruction sets." [Hennessy/Patterson, *Computer Architecture: A Quantitative Approach*, ppg. 472-473]

For multi-core SoCs developed in the Microcontroller Division, a hardware-enforced semaphore implementation as a platform IPS module is used for the following reasons:

- A smaller module size versus a platform reservation monitor for ARM Cortex A5 baseddesigns and it does not introduce any timing-critical paths, for example, the AHB system bus address phase cycle like a reservation monitor
- Provides a more robust implementation with features not possible in a reservation-only approach
- Provides a more generic, architecture-neutral solution, applicable across *all* the processor families supported by the Microcontroller Division

The sole drawback to a hardware-based semaphore module is the limited number of semaphores versus the "infinite" number that can be supported with ARM Cortex A5 based reservation instructions.

24.1.2 Overview

The IPS_Semaphores module provides a platform IPS slave device which implements 16 hardware-enforced gates with the following features:

- Module definition supports 16 hardware-enforced gates in a dual-processor configuration, where cp0 is core processor 0 and cp1 is core processor 1
    - Hardware gates appear as a 16-entry byte-size array with read and write accesses
        - Processors lock gates by writing "processor_number+1" to the appropriate gate and must read back the gate value to verify the lock operation was successful
        - Once locked, the gate is unlocked by a write of zeroes from the locking processor
    - Optional interrupt notification after a failed lock write provides a mechanism to indicate when the gate is unlocked

  - Secure reset mechanisms are supported to clear the contents of individual semaphore gates or notification logic, as well as a clear_all capability
  - Programming model allocates memory space to support up to 8 processors and up to 64 gates
- Memory-mapped IPS platform device
  - Interface to the IPS bus for programming-model accesses
  - Two outputs (one per processor) for interrupt notification of failed lock writes
- Implementation Specifics
  - Module size estimate, based on 130-nm (HiP7A) process technology @ 150 C PVT, is ~2.5 Kgates (nand2_2 equivalents) with no timing issues
  - Implementation includes multiple, independent, but related finite-state machines

A simplified block diagram of the Semaphores module is shown in Figure 24-1. In the diagram, the register blocks named gate0, gate1,…, gate 15 include the finite state machines implementing the semaphore gates plus the interrupt notification logic.

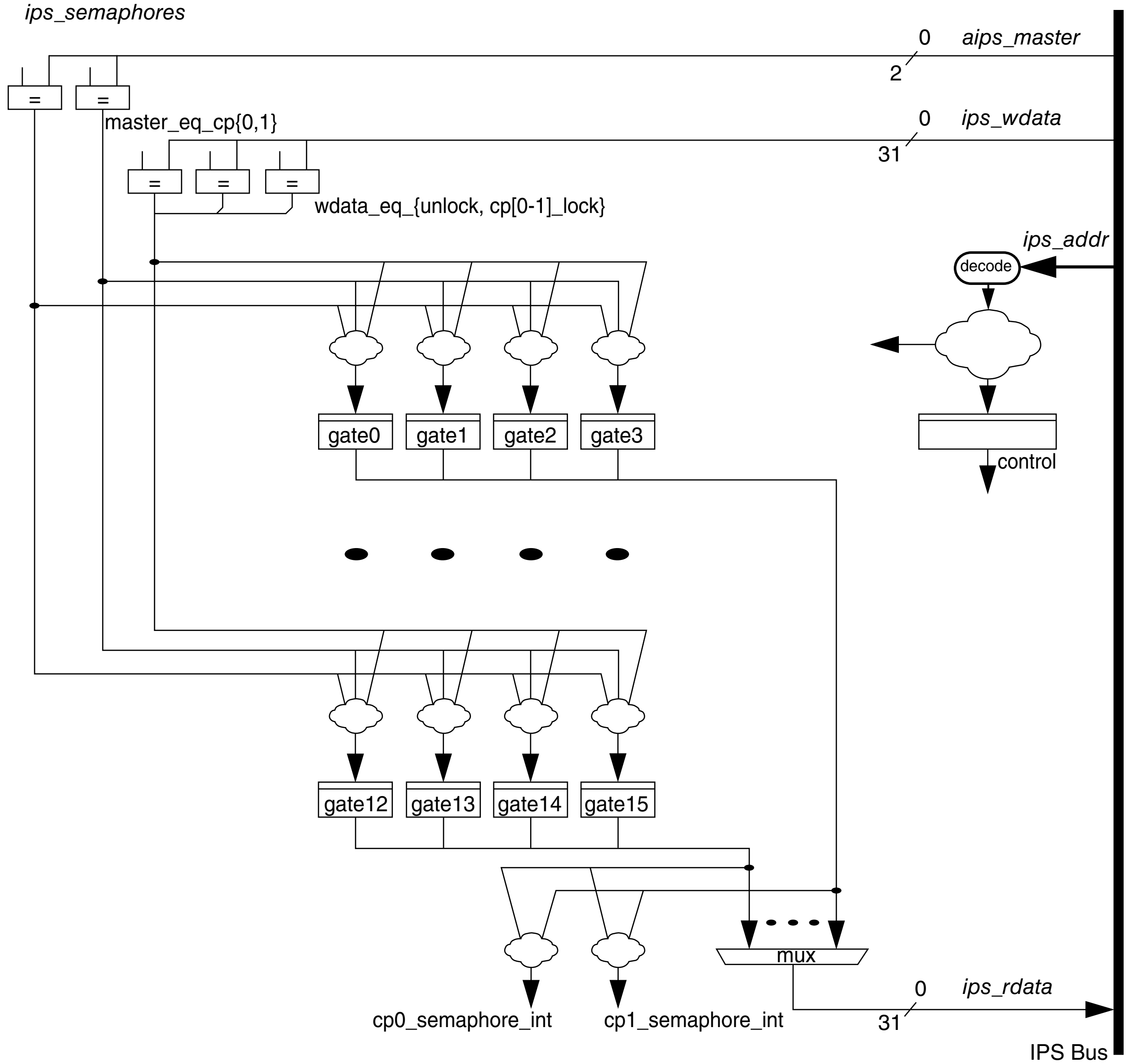


Figure 24-1. IPS_Semaphores Block Diagram

24.1.3 Features

The Semaphores module implements hardware-enforced semaphores as an IPS-mapped slave peripheral device. The feature set includes:

- Support for 16 hardware-enforced gates in a dual-processor configuration
  - Each hardware gate appears as a 3-state, 2-bit state machine, with all 16 gates mapped as a byte-size array
    - 3-state implementation

      if gate = 0b00, then state = unlocked
      if gate = 0b01, then state = locked by processor 0

if gate = 0b10, then state = locked by processor 1

- Uses the bus master number as a reference attribute plus the specified data patterns to validate all write operations
- Once locked, the gate can (and must) be unlocked by a write of zeroes from the locking processor

- Optional interrupt notification after a failed lock write provides a mechanism to indicate when the gate is unlocked
- Secure reset mechanisms are supported to clear the contents of individual gates or notification logic, as well as a clear_all capability

- Memory-mapped IPS slave peripheral platform module
    - Interface to the IPS bus for programming-model accesses
    - Two outputs (one per processor) for interrupt notification of failed lock writes

24.1.4 Modes of Operation

The Semaphores module does not support any special modes of operation. As a slave peripheral memory-mapped device located on the platform's IPS slave bus, it responds based strictly on the memory addresses of the connected bus. The IPS bus is used to access the Semaphores ' programming model.

24.2 External Signal Description

The Semaphores module does not include any external interfaces. The Semaphores ' *internal* interface include an IPS connection for accessing the programming model plus interrupt requests for failed lock processor notification. For additional information on the internal interfaces, see IPS_Semaphores Interfaces.

24.3 Memory map and register definition

The Semaphores module provides an IPS programming model mapped to an SPP-standard on-platform 16 KB space. The description here specifies a dual-core configuration with 16 semaphore gates. All the register names are prefixed with "Sema4" as an abbreviation for the full module name.

The programming model is referenced using 8-, 16- and 32-bit accesses. Reads can use any reference size, while writes are generally restricted to the size of the register. Exceptions to the write size restrictions are detailed in the individual register descriptions. Attempted references using inappropriate access sizes, to undefined (reserved) addresses, or with a non-supported access type (for example, a write to a read-only register) generate an IPS error termination.

Finally, the programming model allocates space for a definition with up to 64 gates and up to 8 processor cores, even though this definition is considerably larger than any currently-planned module implementations. The number of gates and supported processor cores are independent; there is no relationship between these two system variables.

The 16 KB Semaphores programming model map is shown in the following table.

SEMA4 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_D000	Semaphores Gate 0 Register (SEMA4_Gate00)	8	R/W	00h	24.3.1/1086
4001_D001	Semaphores Gate 1 Register (SEMA4_Gate01)	8	R/W	00h	24.3.2/1087
4001_D002	Semaphores Gate 2 Register (SEMA4_Gate02)	8	R/W	00h	24.3.3/1088
4001_D003	Semaphores Gate 3 Register (SEMA4_Gate03)	8	R/W	00h	24.3.4/1089
4001_D004	Semaphores Gate 4 Register (SEMA4_Gate04)	8	R/W	00h	24.3.5/1090
4001_D005	Semaphores Gate 5 Register (SEMA4_Gate05)	8	R/W	00h	24.3.6/1091
4001_D006	Semaphores Gate 6 Register (SEMA4_Gate06)	8	R/W	00h	24.3.7/1092
4001_D007	Semaphores Gate 7 Register (SEMA4_Gate07)	8	R/W	00h	24.3.8/1093
4001_D008	Semaphores Gate 8 Register (SEMA4_Gate08)	8	R/W	00h	24.3.9/1094
4001_D009	Semaphores Gate 9 Register (SEMA4_Gate09)	8	R/W	00h	24.3.10/ 1095
4001_D00A	Semaphores Gate 10 Register (SEMA4_Gate10)	8	R/W	00h	24.3.11/ 1096
4001_D00B	Semaphores Gate 11 Register (SEMA4_Gate11)	8	R/W	00h	24.3.12/ 1097
4001_D00C	Semaphores Gate 12 Register (SEMA4_Gate12)	8	R/W	00h	24.3.13/ 1098
4001_D00D	Semaphores Gate 13 Register (SEMA4_Gate13)	8	R/W	00h	24.3.14/ 1099
4001_D00E	Semaphores Gate 14 Register (SEMA4_Gate14)	8	R/W	00h	24.3.15/ 1100
4001_D00F	Semaphores Gate 15 Register (SEMA4_Gate15)	8	R/W	00h	24.3.16/ 1101
4001_D040	Semaphores Processor n IRQ Notification Enable (SEMA4_CP0INE)	16	R/W	0000h	24.3.17/ 1102
4001_D048	Semaphores Processor n IRQ Notification Enable (SEMA4_CP1INE)	16	R/W	0000h	24.3.17/ 1102

Table continues on the next page...

SEMA4 memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_D080	Semaphores Processor n IRQ Notification (SEMA4_CP0NTF)	16	R	0000h	24.3.18/ 1104
4001_D088	Semaphores Processor n IRQ Notification (SEMA4_CP1NTF)	16	R	0000h	24.3.18/ 1104
4001_D100	Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)	16	R/W	0000h	24.3.19/ 1105
4001_D104	Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)	16	R/W	0000h	24.3.20/ 1107

24.3.1 Semaphores Gate 0 Register (SEMA4_Gate00)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 0h offset = 4001_D000h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate00 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation .

Table continues on the next page...

SEMA4_Gate00 field descriptions (continued)

Field	Description
	NOTE: The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.2 Semaphores Gate 1 Register (SEMA4_Gate01)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 1h offset = 4001_D001h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate01 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free).

Table continues on the next page...

SEMA4_Gate01 field descriptions (continued)

Field	Description
	01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.3 Semaphores Gate 2 Register (SEMA4_Gate02)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 2h offset = 4001_D002h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate02 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.4 Semaphores Gate 3 Register (SEMA4_Gate03)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 3h offset = 4001_D003h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate03 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.5 Semaphores Gate 4 Register (SEMA4_Gate04)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 4h offset = 4001_D004h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate04 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.6 Semaphores Gate 5 Register (SEMA4_Gate05)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 5h offset = 4001_D005h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate05 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.7 Semaphores Gate 6 Register (SEMA4_Gate06)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 6h offset = 4001_D006h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate06 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.8 Semaphores Gate 7 Register (SEMA4_Gate07)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 7h offset = 4001_D007h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate07 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.9 Semaphores Gate 8 Register (SEMA4_Gate08)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 8h offset = 4001_D008h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate08 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.10 Semaphores Gate 9 Register (SEMA4_Gate09)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + 9h offset = 4001_D009h

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate09 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.11 Semaphores Gate 10 Register (SEMA4_Gate10)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + Ah offset = 4001_D00Ah

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate10 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.12 Semaphores Gate 11 Register (SEMA4_Gate11)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + Bh offset = 4001_D00Bh

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate11 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.13 Semaphores Gate 12 Register (SEMA4_Gate12)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + Ch offset = 4001_D00Ch

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate12 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.14 Semaphores Gate 13 Register (SEMA4_Gate13)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + Dh offset = 4001_D00Dh

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate13 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.15 Semaphores Gate 14 Register (SEMA4_Gate14)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + Eh offset = 4001_D00Eh

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate14 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.16 Semaphores Gate 15 Register (SEMA4_Gate15)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 4001_D000h base + Fh offset = 4001_D00Fh

Bit	0	1	2	3	4	5	6	7
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate15 field descriptions

Field	Description
0–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–7 GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

24.3.17 Semaphores Processor n IRQ Notification Enable (SEMA4_CP*n*INE)

The application of a hardware semaphore module provides an opportunity for implementation of helpful system-level features. An example is an optional mechanism to generate a processor interrupt after a failed lock attempt. Recall traditional software gate functions execute a spin-wait loop in an effort to obtain and lock the referenced gate. With this module, the processor that fails in the lock attempt could continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function.

The optional notification interrupt function consists of two registers for each processor: an interrupt notification enable register (SEMA4_CPnINE) and the interrupt request register (SEMA4_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4_CPnINE registers, unimplemented bits read as zeroes, and writes are ignored.

Address: 4001_D000h base + 40h offset + (8d × i), where i=0d to 1d

Bit	0	1	2	3	4	5	6	7
Read / Write	INE0	INE1	INE2	INE3	INE4	INE5	INE6	INE7
Reset	0	0	0	0	0	0	0	0

Bit	8	9	10	11	12	13	14	15
Read / Write	INE8	INE9	INE10	INE11	INE12	INE13	INE14	INE15
Reset	0	0	0	0	0	0	0	0

SEMA4_CP*n*INE field descriptions

Field	Description
0 INE0	Interrupt Request Notification Enable 0. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 0. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
1 INE1	Interrupt Request Notification Enable 1. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 1. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
2 INE2	Interrupt Request Notification Enable 2. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 2. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
3 INE3	Interrupt Request Notification Enable 3. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 3.

Table continues on the next page...

SEMA4_CP*n*INE field descriptions (continued)

Field	Description
	0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
4 INE4	Interrupt Request Notification Enable 4. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 4. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
5 INE5	Interrupt Request Notification Enable 5. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 5. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
6 INE6	Interrupt Request Notification Enable 6. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 6. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
7 INE7	Interrupt Request Notification Enable 7. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 7. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
8 INE8	Interrupt Request Notification Enable 8. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 8. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
9 INE9	Interrupt Request Notification Enable 9. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 9. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
10 INE10	Interrupt Request Notification Enable 10. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 10. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
11 INE11	Interrupt Request Notification Enable 11. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 11. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
12 INE12	Interrupt Request Notification Enable 12. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 12. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
13 INE13	Interrupt Request Notification Enable 13. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 13.

Table continues on the next page...

SEMA4_CP*n*INE field descriptions (continued)

Field	Description
	0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
14 INE14	Interrupt Request Notification Enable 14. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 14. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
15 INE15	Interrupt Request Notification Enable 15. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 15. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.

24.3.18 Semaphores Processor n IRQ Notification (SEMA4_CP*n*NTF)

The Semaphores module optionally allows the processor that fails in the lock attempt to continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function rather than simply execute in a spin-wait loop.

The optional notification interrupt mechanism consists of two registers for each processor: an interrupt notification enable register (SEMA4_CPnINE) and the read-only notification interrupt request register (SEMA4_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4_CPnNTF registers, unimplemented bits read as zeroes.

The notification interrupt is generated via a unique finite state machine, one per hardware gate. This machine operates in the following manner:

1. When an attempted lock fails, the FSM enters a first state where it waits until the gate is unlocked.
2. Once unlocked, the FSM enters a second state where it generates an interrupt request to the “failed lock” processor.
3. When the “failed lock” processor succeeds in locking the gate, the IRQ is automatically negated and the FSM returns to the idle state. However, if the other processor again locks the gate, the FSM returns to the first state, negates the interrupt request, and then waits for the gate to be unlocked (again).

The notification interrupt request is implemented in a 3-bit, 5-state machine, where tw specific states are encoded and program-visible as SEMA4_CP0NTF[GNn] and SEMA4_CP1NTF[GNn].

Address: 4001_D000h base + 80h offset + (8d × i), where i=0d to 1d

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Read Write	GNn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEMA4_CP*n*NTF field descriptions

Field	Description
0–15 GNn	Gate n Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate n. For more details, see SEMA4_CPnNTF Operation .

24.3.19 Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the Semaphores module implements a "secure" reset mechanism which allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4_RSTGT memory location. The most significant byte (SEMA4_RSTGT[RSTGDP]) must be 0xe2; the least significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4_RSTGT location. For this write, the upper byte (SEMA4_RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1d) and the lower byte (SEMA4_RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or that all gates are cleared.
3. Reads of the SEMA4_RSTGT location return information on the 2-bit state machine (SEMA4_RSTGT[RSTGSM]) which implements this function, the bus master performing the reset (SEMA4_RSTGT[RSTGMS]) and the gate number(s) last cleared (SEMA4_RSTGT[RSTGTN]). Reads of the SEMA4_RSTGT register do not affect the secure reset finite state machine in any manner.

Address: 4001_D000h base + 100h offset = 4001_D100h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Read Write	RSTGSM_RSTGMS_RSTGDP								RSTGTN							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEMA4_RSTGT field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>0–7
RSTGSM_
RSTGMS_
RSTGDP</td><td>

NOTE: This field contains subfields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details.

<table>
<tr><th>Access</th><th>Sub-Field</th><th>Description</th></tr>
<tr><td rowspan="4">Read-Only</td><td>0-1
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>2-3
RSTGSM</td><td>Reset Gate Finite State Machine. Reads of the SEMA4_RSTGT register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:
<table>
<tr><td>00</td><td>Idle, waiting for the first data pattern write.</td></tr>
<tr><td>01</td><td>Waiting for the second data pattern write.</td></tr>
<tr><td>10</td><td>The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note that the RSTGSM = 0b10 state is valid for only a single machine cycle, so it is impossible for a read to return this value</td></tr>
<tr><td>11</td><td>This state encoding is never used and therefore reserved.</td></tr>
</table></td></tr>
<tr><td>4
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>5-7
RSTGMS</td><td>Reset Gate Bus Master. This 3-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip configuration chapter for this information.</td></tr>
<tr><td>Write-Only</td><td>0-7
RSTGDP</td><td>Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xe2 while the second write requires RSTGDP = 0x1d.</td></tr>
</table>
</td></tr>
<tr><td>8–15
RSTGTN</td><td>Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write.

If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates. The corresponding secure IRQ notification state machine(s) are also reset.</td></tr>
</table>

24.3.20 Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)

As with the case of the secure reset function and the hardware gates, it is recognized that system operation may require a reset function to re-initialize the state of the IRQ notification logic without requiring a system-level reset.

To support this special notification reset requirement, the Semaphores module implements a "secure" reset mechanism which allows an IRQ notification (or all the notifications) to be initialized by following a specific dual-write access pattern. When successful, the specified IRQ notification state machine(s) are reset. Using a technique similar to that required for the servicing of a software watchdog timer, the secure reset mechanism requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the IRQ notification(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4_RSTNTF memory location. The most significant byte (SEMA4_RSTNTF[RSTNDP]) must be 0x47; the least significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4_RSTNTF location. For this write, the upper byte (SEMA4_RSTNTF[RSTNDP]) is the logical complement of the first data pattern (0xb8) and the lower byte (SEMA4_RSTNTF[RSTNTN]) specifies the notification(s) to be reset. This field can specify a single notification be cleared, or that all notifications are cleared.
3. Reads of the SEMA4_RSTNTF location return information on the 2-bit state machine (SEMA4_RSTNTF[RSTNSM]) which implements this function, the bus master performing the reset (SEMA4_RSTNTF[RSTNMS]) and the notification number(s) last cleared (SEMA4_RSTNTF[RSTNTN]). Reads of the SEMA4_RSTNTF register do not affect the secure reset finite state machine in any manner.

Address: 4001_D000h base + 104h offset = 4001_D104h

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Read Write	RSTNSM_RSTNMS_RSTNDP								RSTNTN							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEMA4_RSTNTF field descriptions

Field	Description
0–7 RSTNSM_ RSTNMS_ RSTNDP	**NOTE:** This field contains subfields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details.

Table continues on the next page...

SEMA4_RSTNTF field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Access</th><th>Sub-Field</th><th>Description</th></tr>
<tr><td rowspan="4">Read-Only</td><td>0-1
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>2-3
RSTNSM</td><td>Reset Notification Finite State Machine. Reads of the SEMA4_RSTNTF register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:
<table>
<tr><td>00</td><td>Idle, waiting for the first data pattern write.</td></tr>
<tr><td>01</td><td>Waiting for the second data pattern write.</td></tr>
<tr><td>10</td><td>The 2-write sequence has completed. Generate the specified notification reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note the RSTNSM = 10 state is valid for only a single machine cycle, so it is impossible for a read to return this value.</td></tr>
<tr><td>11</td><td>This state encoding is never used and therefore reserved..</td></tr>
</table></td></tr>
<tr><td>4
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>5-7
RSTNMS</td><td>Reset Notification Bus Master. This 3-bit read-only field records the logical number of the bus master performing the notification reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip configuration chapter for this information.</td></tr>
<tr><td>Write-Only</td><td>0-7
RSTNDP</td><td>Reset Notification Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the notification reset mechanism. For the first write, RSTNDP = 0x47 while the second write requires RSTNDP = 0xb8.</td></tr>
</table>
</td></tr>
<tr><td>8–15
RSTNTN</td><td>Reset Notification Number. This 8-bit field specifies the specific IRQ notification state machine to be reset. This field is updated by the second write.

If RSTNTN < 64, then reset the single IRQ notification machine defined by RSTNTN, else reset all the notifications.</td></tr>
</table>

24.4 Functional Description

In this section, the functional operation of the Semaphores module, specifically the state machines of the SEMA4_GATEn and SEMA4_CPnNTF registers are detailed.

24.4.1 SEMA4_GATEn Operation

Recall each of the SEMA4_GATEn registers implements a 2-bit, 3-state machine. The state transitions for each gate are shown in the following figure.

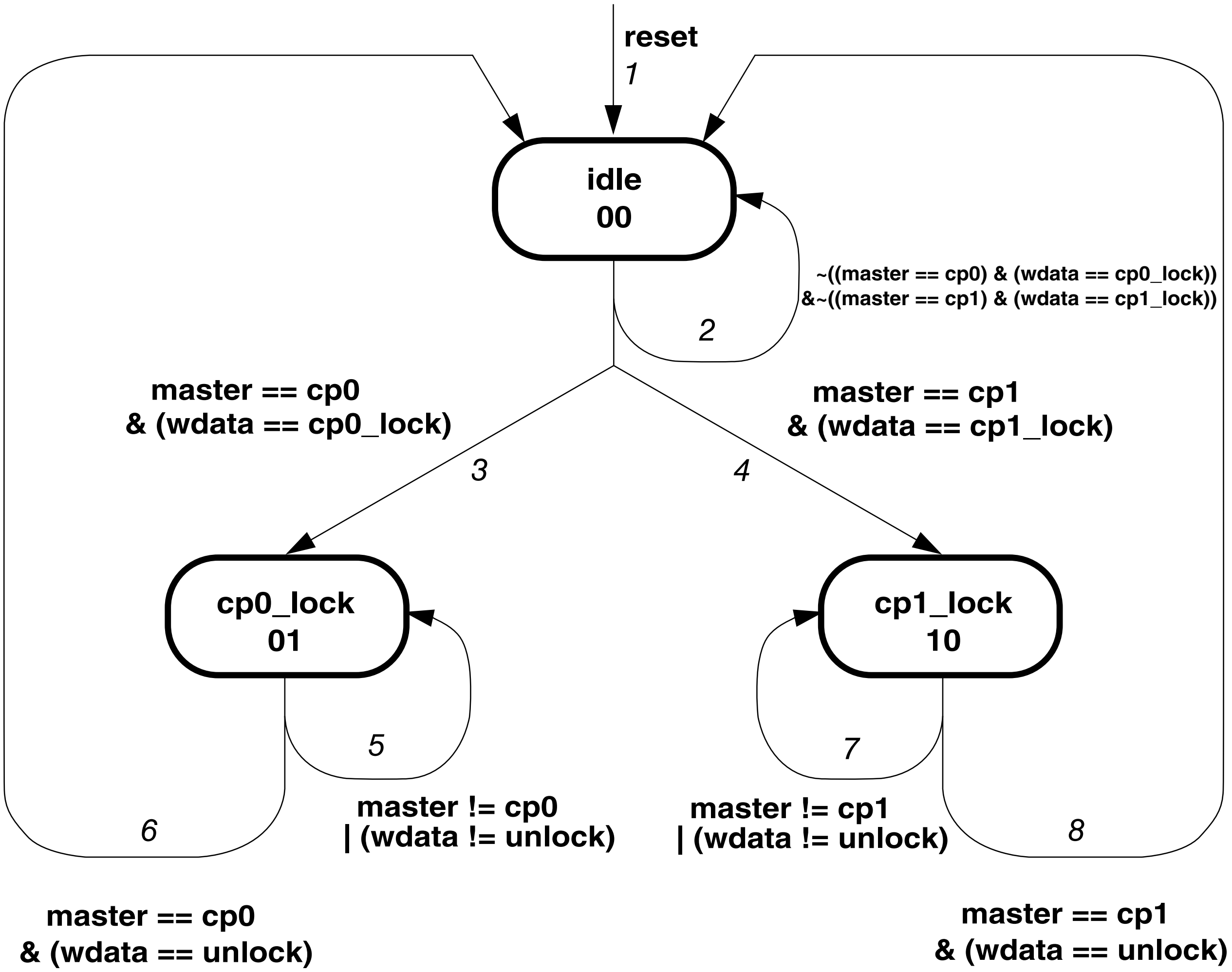


Figure 24-26. SEMA4_GATEn State Machine

The bus master number is used to identify core processor 0 (cp0) or core processor 1 (cp1). The Standard (or Reduced) Product Platform passes the AHB bus master number (`hmaster[2:0]`) through the AIPS (or AIPS-Lite) controller and drives an `aips_master[2:0]` output to the Semaphores module as an IPS sideband signal.

The state transitions for SEMA4_GATEn are defined in the following table.

Table 24-26. SEMA4_GATEn State Transitions

Current State	Next State	Transition	Description
–	idle	1	Any reset, whether a system reset or an individual gate reset, unconditionally forces the gate into the idle state.

Table continues on the next page...

Table 24-26. SEMA4_GATEn State Transitions (continued)

Current State	Next State	Transition	Description
idle	idle	2	Unless a write of the appropriate lock value from the corresponding processor occurs, the gate remains in the idle state.
idle	cp0_lock	3	When a write of the "cp0_lock" data value is initiated by processor 0, the gate transitions into the cp0_lock state.
idle	cp1_lock	4	When a write of the "cp1_lock" value is initiated by processor 1, the gate transitions into the cp1_lock state.
cp0_lock	cp0_lock	5	Once in this state, the gate remains here if any attempted write is not from cp0 with the unlock data value.
cp0_lock	idle	6	The gate returns to the idle (unlocked) state once a write from cp0 with the unlock data value occurs.
cp1_lock	cp1_lock	7	Once in this state, the gate remains here if any attempted write is not from cp1 with the unlock data value.
cp1_lock	idle	8	The gate returns to the idle (unlocked) state once a write from cp1 with the unlock data value occurs.

24.4.2 SEMA4_CPnNTF Operation

The failed lock write notification interrupt request is implemented in a 3-bit, 5-state machine which records failed lock attempts and transitions based on gate locking and unlocking. Two specific states are encoded and program-visible as SEMA4_CP0NTF[GNn] and SEMA4_CP1NTF[GNn]. See the following figure.

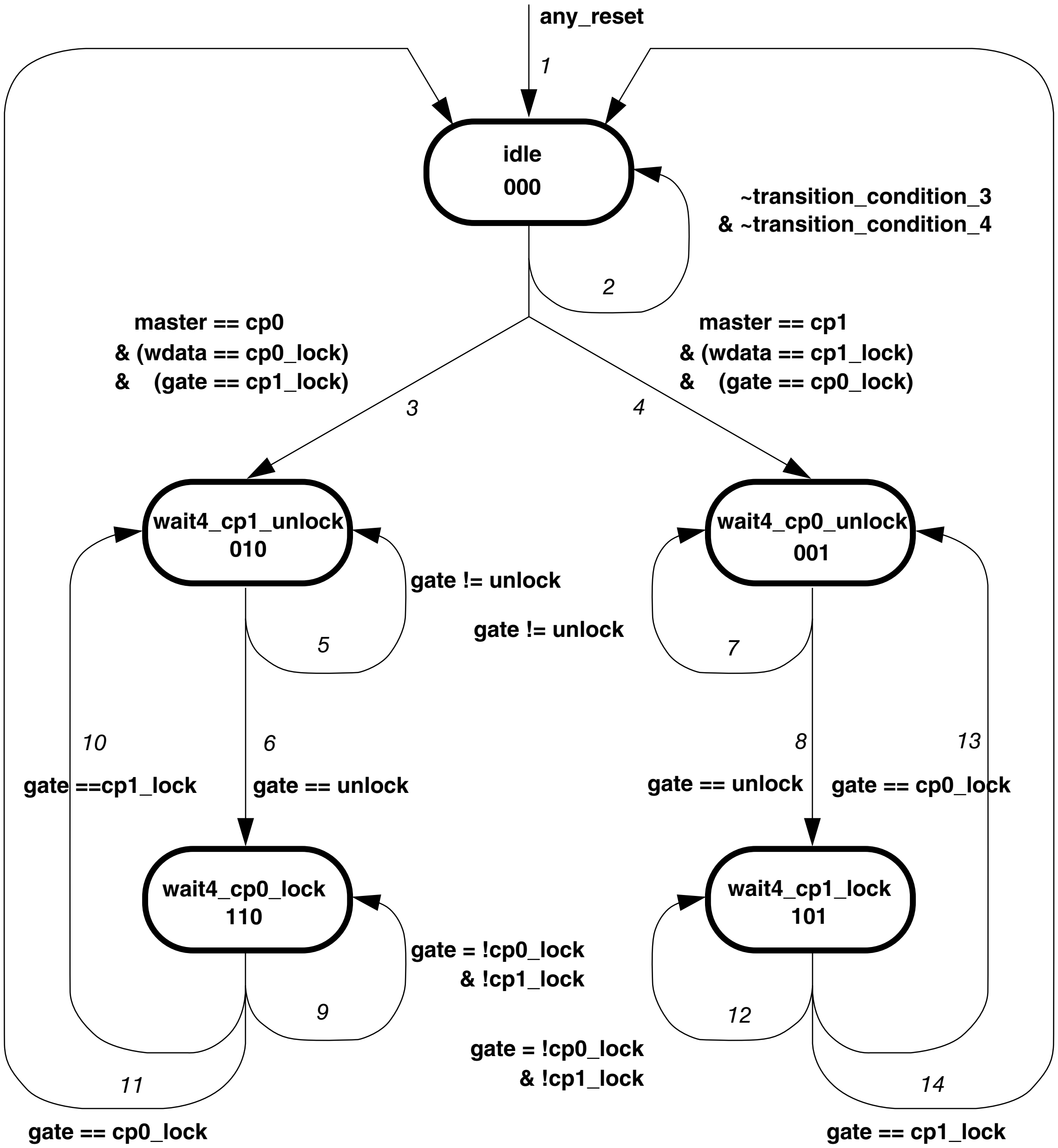


Figure 24-27. IRQ Notification State Machine

The state transitions of the IRQ notification function are defined in the following. Specific states of this machine are program-visible as the SEMA4_CPnNTF registers. In particular, two states are program-visible:

```
if state = wait4_cp0_lock (0b110) // generate cp0_semaphore_int if properly enabled
                then SEMA4_CP0NTF[GNn] = 1; else SEMA4_CP0NTF[GNn] = 0
if state = wait4_cp1_lock (0b101) // generate cp1_semaphore_int if properly enabled
                then SEMA4_CP1NTF[GNn] = 1; else SEMA4_CP1NTF[GNn] = 0
```

Table 24-27. IRQ Notification State Transitions

Current State	Next State	Transition	Description
–	idle	1	Any reset, including a system reset or an individual notification or secure gate reset, unconditionally forces the machine into the idle state.
idle	idle	2	Unless a write of the appropriate lock value from the corresponding processor to an already-locked gate occurs, the machine remains in the idle state.
idle	wait4_cp1_unlock	3	When a write of the "cp0_lock" data value is initiated by processor 0 but the gate is already locked by cp1, the machine transitions into this state, where it waits for cp1 to unlock the gate.
idle	wait4_cp0_unlock	4	When a write of the "cp1_lock" data value is initiated by processor 1but the gate is already locked by cp0, the machine transitions into this state, where it waits for cp0 to unlock the gate
wait4_cp1_unlock	wait4_cp1_unlock	5	Once in this state, the machine remains here until the gate is unlocked.
wait4_cp1_unlock	wait4_cp0_lock	6	From this state, the machine transitions into the next state, waiting for cp0 to lock the gate, once it has been unlocked.
wait4_cp0_unlock	wait4_cp0_unlock	7	Once in this state, the machine remains here until the gate is unlocked.
wait4_cp0_unlock	wait4_cp1_lock	8	From this state, the machine transitions into the next state, waiting for cp1 to lock the gate, once it has been unlocked.
wait4_cp0_lock	wait4_cp0_lock	9	In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 0 or the gate is again locked by processor 1.
wait4_cp0_lock	wait4_cp1_unlock	10	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 1. With this transition, the notification interrupt request is negated.
wait4_cp0_lock	idle	11	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 0. With this transition, the notification interrupt request is negated.
wait4_cp1_lock	wait4_cp1_lock	12	In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 1or the gate is again locked by processor 0.
wait4_cp1_lock	wait4_cp0_unlock	13	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 0. With this transition, the notification interrupt request is negated.
wait4_cp1_lock	idle	14	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 1. With this transition, the notification interrupt request is negated.

The Semaphores module generates two interrupt request output signals, one per processor, combining the SEMA4_CPnINE and SEMA4_CPnNTF registers, where the boolean equations are:

```
cp0_semaphore_int
    =    sema4_cp0ine[ine0]    &    sema4_cp0ntf[gn0]
    |    sema4_cp0ine[ine1]    &    sema4_cp0ntf[gn1]
    |    sema4_cp0ine[ine2]    &    sema4_cp0ntf[gn2]
    ...
    |    sema4_cp0ine[ine15]    &    sema4_cp0ntf[gn15]
cp1_semaphore_int
    =    sema4_cp1ine[ine0]    &    sema4_cp1ntf[gn0]
    |    sema4_cp1ine[ine1]    &    sema4_cp1ntf[gn1]
    |    sema4_cp1ine[ine2]    &    sema4_cp1ntf[gn2]
    ...
    |    sema4_cp1ine[ine15]    &    sema4_cp1ntf[gn15]
```

24.5 Initialization Information

The reset state of the IPS_Semaphores module allows it to begin operation without the need for any further initialization. All the internal state machines are cleared by any reset event, allowing the module to immediately begin operation.

24.6 Application Information

In an operational multi-core system, most interactions involving the Semaphores module involves reads and writes to the SEMA4_GATEn registers for implementation of the hardware-enforced software gate functions. Typical code segments for gate functions perform the following operations:

- To lock (close) a gate
  - The processor performs a byte write of "logical_processor_number + 1" to gate[i]
  - The processor reads back gate[i] and checks for a value of "logical_processor_number + 1"

If the compare indicates the expected value

then the gate is locked; proceed with the protected code segment

else

lock operation failed;

repeat process beginning with byte write to gate[i] in spin-wait loop, or

proceed with another execution path and wait for failed lock interrupt notification

A simple C-language example of a `gateLock` function is shown in the following figure. This function follows the Hennessy/Patterson example described in Multi-Core Programming 101: Software Gates.

```
#define UNLOCK      0
#define CP0_LOCK    1
#define CP2_LOCK    2

void gateLock (n)
int   n;                              /* gate number to lock */
{
  int i;
  int current_value;
  int locked_value;

  i = processor_number();  /* obtain logical CPU number */

  if (i == 0)
    locked_value = CP0_LOCK;
  else
    locked_value = CP1_LOCK;

  /* read the current value of the gate and wait until the state == UNLOCK */
  do {
     current_value = gate[n];
  } while (current_value != UNLOCK);

  /* the current value of the gate == UNLOCK. attempt to lock the gate for this
     processor. spin-wait in this loop until gate ownership is obtained */
  do {
    gate[n] = locked_value;  /* write gate with processor_number + 1 */
    current_value = gate[n]; /* read gate to verify ownership was obtained */
  } while (current_value != locked_value);
```

Figure 24-28. Sample gateLock Function

- To unlock (open) a gate
  - After completing the protected code segment, the locking processor performs a byte write of zeroes to gate[i], opening (unlocking) the gate

A few comments on the logical CPU number are appropriate. In this example, a reference to `processor_number()` is used to retrieve this hardware configuration value. Typically, the logical processor numbers are defined by a hardwired input vector to the individual cores. The exact method for accessing the logical processor number varies by architecture. For PowerPC cores, there is a processor ID register (PIR) which is SPR 286 and contains this value. A single instruction can be used to move the contents of the PIR into a general-purpose register: **`mfspr rx,286`** where rx is the destination GPRn. Other architectures may support a specific instruction to move the contents of the logical processor number into a general-purpose register, e.g., **`rdcpn rx`** for a "read CPU number" instruction.

If the optional failed lock IRQ notification mechanisms are used, then accesses to the related registers (SEMA4_CPnINE, SEMA4_ CPnNTF) are required. Note that there is no required negation of the failed lock write notification interrupt as the request is automatically negated by the Semaphores module once the gate has been successfully locked by the "failing" processor.

Finally, in the event a system state requires a software-controlled reset of a gate or IRQ notification register(s), accesses to the secure reset control registers (SEMA4_RSTGT, SEMA4_RSTNTF) are required. For these situations, it is recommended that the appropriate IRQ notification enable(s) (SEMA4_CPnINE) bits be disabled *before* initiating the secure reset 2-write sequence to avoid any race conditions involving spurious notification interrupt requests.

Chapter 25
External Watchdog Monitor (EWM)

25.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The watchdog is generally used to monitor the flow and execution of embedded software within an MCU. The watchdog consists of a counter that if allowed to overflow, forces an internal reset (asynchronous) to all on-chip peripherals and optionally assert the $\overline{\text{RESET}}$ pin to reset external devices/circuits. The overflow of the watchdog counter must not occur if the software code works well and services the watchdog to re-start the actual counter.

For safety, a redundant watchdog system, External Watchdog Monitor (EWM), is designed to monitor external circuits, as well as the MCU software flow. This provides a back-up mechanism to the internal watchdog that resets the MCU's CPU and peripherals.

The EWM differs from the internal watchdog in that it does not reset the MCU's CPU and peripherals. The EWM if allowed to time-out, provides an independent $\overline{\text{EWM\_out}}$ pin that when asserted resets or places an external circuit into a safe mode. The CPU resets the EWM counter that is logically ANDed with an external digital input pin. This pin allows an external circuit to influence the reset_out signal.

25.1.1 Features

Features of EWM module include:

- Independent LPO clock source
- Programmable time-out period specified in terms of number of EWM LPO clock cycles.

- Windowed refresh option
  - Provides robust check that program flow is faster than expected.
  - Programmable window.
  - Refresh outside window leads to assertion of $\overline{\text{EWM\_out}}$.
- Robust refresh mechanism
  - Write values of 0xB4 and 0x2C to EWM Refresh Register within 63 (*EWM_service_time*) peripheral bus clock cycles.
- One output port, $\overline{\text{EWM\_out}}$, when asserted is used to reset or place the external circuit into safe mode.
- One Input port, EWM_in, allows an external circuit to control the $\overline{\text{EWM\_out}}$ signal.

25.1.2 Modes of Operation

This section describes the module's operating modes.

25.1.2.1 Stop Mode

When the EWM is in stop mode, the CPU services to the EWM cannot occur. On entry to stop mode, the EWM's counter freezes.

There are two possible ways to exit from Stop mode:

- On exit from stop mode through a reset, the EWM remains disabled.
- On exit from stop mode by an interrupt, the EWM is re-enabled, and the counter continues to be clocked from the same value prior to entry to stop mode.

Note the following if the EWM enters the stop mode during CPU service mechanism: At the exit from stop mode by an interrupt, refresh mechanism state machine starts from the previous state which means, if first service command is written correctly and EWM enters the stop mode immediately, the next command has to be written within the next 63 (*EWM_service_time*) peripheral bus clocks after exiting from stop mode. User must mask all interrupts prior to executing EWM service instructions.

25.1.2.2 Wait Mode

The EWM module treats the stop and wait modes as the same. EWM functionality remains the same in both of these modes.

25.1.2.3 Debug Mode

Entry to debug mode has no effect on the EWM.

- If the EWM is enabled prior to entry of debug mode, it remains enabled.
- If the EWM is disabled prior to entry of debug mode, it remains disabled.

25.1.3 Block Diagram

This figure shows the EWM block diagram.

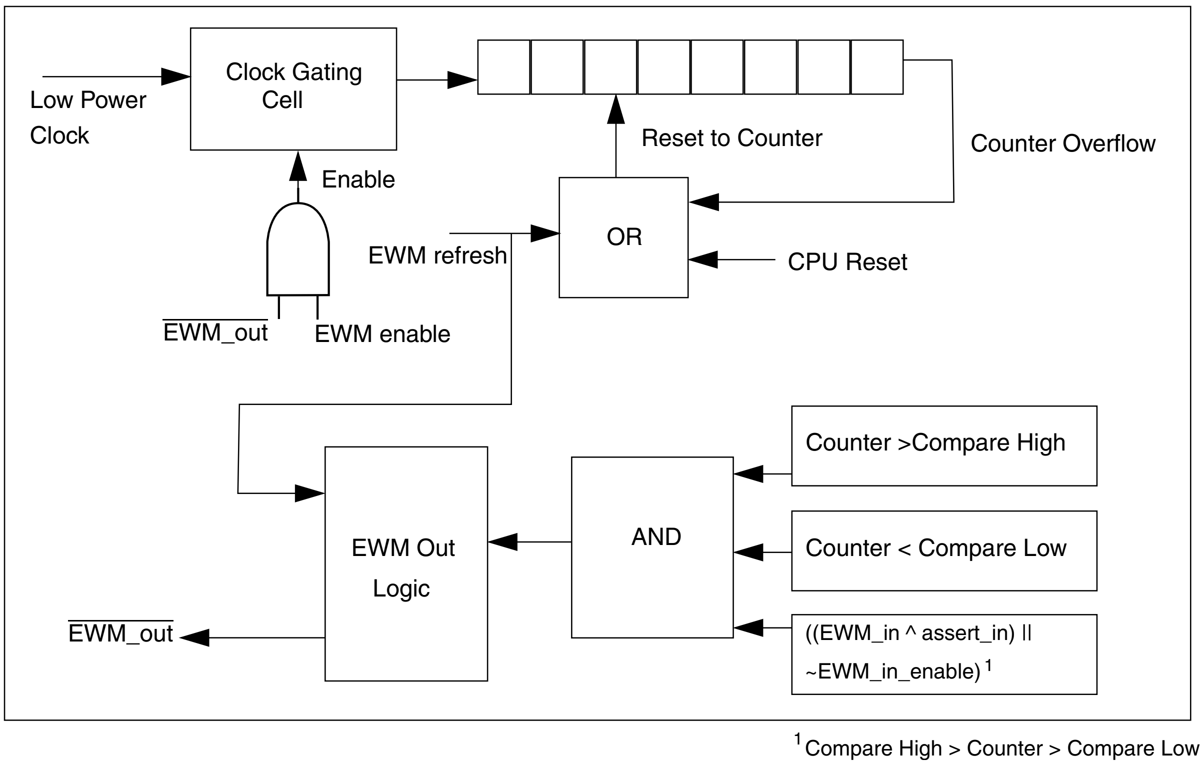


Figure 25-1. EWM Block Diagram

25.2 EWM Signal Descriptions

The EWM has two external signals and internal options for the counter clock sources, as shown in the following table.

Table 25-1. EWM Signal Descriptions

Signal	Description	I/O
EWM_in	EWM input for safety status of external safety circuits. The polarity of EWM_in is programmable using the EWM_CTRL[ASSIN] bit. The default polarity is active-low.	I
$\overline{\text{EWM\_out}}$	EWM reset out signal	O
lpo_clk[3:0]	Low power clock sources for running counter All four lpo_clk clocks are SOSC clocks.	I

25.3 Memory Map/Register Definition

This section contains the module memory map and registers.

EWM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_5000	Control Register (EWM_CTRL)	8	R/W	00h	25.3.1/1120
4006_5001	Service Register (EWM_SERV)	8	W (always reads 0)	00h	25.3.2/1121
4006_5002	Compare Low Register (EWM_CMPL)	8	R/W	00h	25.3.3/1122
4006_5003	Compare High Register (EWM_CMPH)	8	R/W	FFh	25.3.4/1122
4006_5004	Clock Control Register (EWM_CLKCTRL)	8	R/W	00h	25.3.5/1123
4006_5005	Clock Prescaler Register (EWM_CLKPRESCALER)	8	R/W	00h	25.3.6/1123

25.3.1 Control Register (EWM_CTRL)

The CTRL register is cleared by any reset.

NOTE

INEN, ASSIN and EWMEN bits can be written once after a CPU reset. Modifying these bits more than once, generates a bus transfer error.

Address: 4006_5000h base + 0h offset = 4006_5000h

Bit	7	6	5	4	3	2	1	0
Read	0				INTEN	INEN	ASSIN	EWMEN
Write								
Reset	0	0	0	0	0	0	0	0

EWM_CTRL field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 INTEN	Interrupt Enable. This bit when set and EWM_out is asserted, an interrupt request is generated. To de-assert interrupt request, user should clear this bit by writing 0.
2 INEN	Input Enable. This bit when set, enables the EWM_in port.
1 ASSIN	EWM_in's Assertion State Select. Default assert state of the EWM_in signal is logic zero. Setting ASSIN bit inverts the assert state to a logic one.
0 EWMEN	EWM enable. This bit when set, enables the EWM module. This resets the EWM counter to zero and deasserts the EWM_out signal. Clearing EWMEN bit disables the EWM, and therefore it cannot be enabled until a reset occurs, due to the write-once nature of this bit.

25.3.2 Service Register (EWM_SERV)

The SERV register provides the interface from the CPU to the EWM module. It is write-only and reads of this register return zero.

Address: 4006_5000h base + 1h offset = 4006_5001h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write	SERVICE							
Reset	0	0	0	0	0	0	0	0

EWM_SERV field descriptions

Field	Description
7–0 SERVICE	The EWM service mechanism requires the CPU to write two values to the SERV register: a first data byte of 0xB4, followed by a second data byte of 0x2C. The EWM service is illegal if either of the following conditions is true. • The first or second data byte is not written correctly. • The second data byte is not written within a fixed number of peripheral bus cycles of the first data byte. This fixed number of cycles is called *EWM_service_time*.

25.3.3 Compare Low Register (EWM_CMPL)

The CMPL register is reset to zero after a CPU reset. This provides no minimum time for the CPU to service the EWM counter.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

Address: 4006_5000h base + 2h offset = 4006_5002h

Bit	7	6	5	4	3	2	1	0
Read Write	COMPAREL							
Reset	0	0	0	0	0	0	0	0

EWM_CMPL field descriptions

Field	Description
7–0 COMPAREL	To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) minimum service time is required.

25.3.4 Compare High Register (EWM_CMPH)

The CMPH register is reset to 0xFF after a CPU reset. This provides a maximum of 256 clocks time, for the CPU to service the EWM counter.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

NOTE

The valid values for CMPH are up to 0xFE because the EWM counter never expires when CMPH = 0xFF. The expiration happens only if EWM counter is greater than CMPH.

Address: 4006_5000h base + 3h offset = 4006_5003h

Bit	7	6	5	4	3	2	1	0
Read Write	COMPAREH							
Reset	1	1	1	1	1	1	1	1

EWM_CMPH field descriptions

Field	Description
7–0 COMPAREH	To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) maximum service time is required.

25.3.5 Clock Control Register (EWM_CLKCTRL)

This CLKCTRL register is reset to 0x00 after a CPU reset.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

NOTE

User should select the required low power clock before enabling the EWM.

Address: 4006_5000h base + 4h offset = 4006_5004h

Bit	7	6	5	4	3	2	1	0
Read	0						CLKSEL	
Write								
Reset	0	0	0	0	0	0	0	0

EWM_CLKCTRL field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 CLKSEL	EWM has 4 possible low power clock sources for running EWM counter. One of the clock source can be selected by writing into this field. • 00 - lpo_clk[0] will be selected for running EWM counter. • 01 - lpo_clk[1] will be selected for running EWM counter. • 10 - lpo_clk[2] will be selected for running EWM counter. • 11 - lpo_clk[3] will be selected for running EWM counter.

25.3.6 Clock Prescaler Register (EWM_CLKPRESCALER)

This CLKPRESCALER register is reset to 0x00 after a CPU reset.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

NOTE

Write the required prescaler value before enabling the EWM.

NOTE

The implementation of this register is chip-specific. See the Chip Configuration details.

Address: 4006_5000h base + 5h offset = 4006_5005h

Bit	7	6	5	4	3	2	1	0
Read Write	CLK_DIV							
Reset	0	0	0	0	0	0	0	0

EWM_CLKPRESCALER field descriptions

Field	Description
7–0 CLK_DIV	Selected low power source for running the EWM counter can be prescaled as below. • Prescaled clock frequency = low power clock source frequency/ (1+ CLK_DIV)

25.4 Functional Description

The following sections describe functional details of the EWM module.

25.4.1 The $\overline{\text{EWM\_out}}$ Signal

The $\overline{\text{EWM\_out}}$ is a digital output signal used to gate an external circuit (application specific) that controls critical safety functions. For example, the $\overline{\text{EWM\_out}}$ could be connected to the high voltage transistors circuits that control an AC motor in a large appliance.

The $\overline{\text{EWM\_out}}$ signal remains deasserted when the EWM is being regularly serviced by the CPU within the programmable service window, indicating that the application code is executed as expected.

The $\overline{\text{EWM\_out}}$ signal is asserted in any of the following conditions:

- Servicing the EWM when the counter value is less than CMPL value.
- If the EWM counter value reaches the CMPH value, and no EWM service has occurred.
- Servicing the EWM when the counter value is more than CMPL and less than CMPH values and EWM_in signal is asserted.

- If functionality of EWM_in pin is enabled and EWM_in pin is asserted while servicing the EWM.
- After any reset (by the virtue of the external pull-down mechanism on the EWM_out pin)

On a normal reset, the EWM_out is asserted. To deassert the EWM_out, set EWMEN bit in the CTRL register to enable the EWM.

If the EWM_out signal shares its pad with a digital I/O pin, on reset this actual pad defers to being an input signal. It takes the EWM_out output condition only after you enable the EWM by the EWMEN bit in the CTRL register.

When the EWM_out pin is asserted, it can only be deasserted by forcing a MCU reset.

Note

EWM_out pad must be in pull down state when EWM functionality is used and when EWM is under Reset.

25.4.2 The EWM_in Signal

The EWM_in is a digital input signal that allows an external circuit to control the EWM_out signal. For example, in the application, an external circuit monitors a critical safety function, and if there is fault with this circuit's behavior, it can then actively initiate the EWM_out signal that controls the gating circuit.

The EWM_in signal is ignored if the EWM is disabled, or if INEN bit of CTRL register is cleared, as after any reset.

On enabling the EWM (setting the CTRL[EWMEN] bit) and enabling EWM_in functionality (setting the CTRL[INEN] bit), the EWM_in signal must be in the deasserted state prior to the CPU servicing the EWM. This ensures that the EWM_out stays in the deasserted state; otherwise, the EWM_out pin is asserted.

Note

You must update the CMPH and CMPL registers prior to enabling the EWM. After enabling the EWM, the counter resets to zero, therefore providing a reasonable time after a power-on reset for the external monitoring circuit to stabilize and ensure that the EWM_in pin is deasserted.

25.4.3 EWM Counter

It is an 8-bit ripple counter fed from a clock source that is independent of the peripheral bus clock source. As the preferred time-out is between 1 ms and 100 ms the actual clock source should be in the kHz range.

The counter is reset to zero, after a CPU reset, or a EWM refresh cycle. The counter value is not accessible to the CPU.

25.4.4 EWM Compare Registers

The compare registers CMPL and CMPH are write-once after a CPU reset and cannot be modified until another CPU reset occurs.

The EWM compare registers are used to create a service window, which is used by the CPU to service/refresh the EWM module.

- If the CPU services the EWM when the counter value lies between CMPL value and CMPH value, the counter is reset to zero. This is a legal service operation.
- If the CPU executes a EWM service/refresh action outside the legal service window, $\overline{\text{EWM\_out}}$ is asserted.

It is illegal to program CMPL and CMPH with same value. In this case, as soon as counter reaches (CMPL + 1), $\overline{\text{EWM\_out}}$ is asserted.

25.4.5 EWM Refresh Mechanism

Other than the initial configuration of the EWM, the CPU can only access the EWM by the EWM Service Register. The CPU must access the EWM service register with correct write of unique data within the windowed time frame as determined by the CMPL and CMPH registers. Therefore, three possible conditions can occur:

Table 25-9. EWM Refresh Mechanisms

Condition	Mechanism
A unique EWM service occurs when CMPL < Counter < CMPH.	The software behaves as expected and the counter of the EWM is reset to zero, and $\overline{\text{EWM\_out}}$ pin remains in the deasserted state. **Note:** EWM_in pin is also assumed to be in the deasserted state.
A unique EWM service occurs when Counter < CMPL	The software services the EWM and therefore resets the counter to zero and asserts the $\overline{\text{EWM\_out}}$ pin (irrespective of the EWM_in pin). The $\overline{\text{EWM\_out}}$ pin is expected to gate critical safety circuits.

Table continues on the next page...

Table 25-9. EWM Refresh Mechanisms (continued)

Condition	Mechanism
Counter value reaches CMPH prior to a unique EWM service	The counter value reaches the CMPH value and no service of the EWM resets the counter to zero and assert the $\overline{\text{EWM\_out}}$ pin (irrespective of the EWM_in pin). The $\overline{\text{EWM\_out}}$ pin is expected to gate critical safety circuits.

Any illegal service on EWM has no effect on $\overline{\text{EWM\_out}}$.

25.4.6 EWM Interrupt

When $\overline{\text{EWM\_out}}$ is asserted, an interrupt request is generated to indicate the assertion of the EWM reset out signal. This interrupt is enabled when CTRL[INTEN] is set. Clearing this bit clears the interrupt request but does not affect $\overline{\text{EWM\_out}}$. The $\overline{\text{EWM\_out}}$ signal can be deasserted only by forcing a system reset.

25.4.7 Selecting the EWM counter clock

There are four possible low power clock sources for the EWM counter. Select one of the available clock sources by programming CLKCTRL[CLKSEL].

25.4.8 Counter clock prescaler

The EWM counter clock source can be prescaled by a clock divider, by programming CLKPRESCALER[CLK_DIV]. This divided clock is used to run the EWM counter.

NOTE

The divided clock used to run the EWM counter must be no more than half the frequency of the bus clock.

Chapter 26
Watchdog Timer (WDOG)

26.1 Overview

The Watchdog Timer (WDOG) protects against system failures by providing a method by which to escape from unexpected events or programming errors.

Once the WDOG is activated, it must be serviced by the software on a periodic basis. If servicing does not take place, the timer times out. Upon timeout, the WDOG asserts the internal system reset signal, $\overline{\text{wdog\_rst}}$, to the System Reset Controller (SRC).

Flow diagrams for the timeout counter shown in .

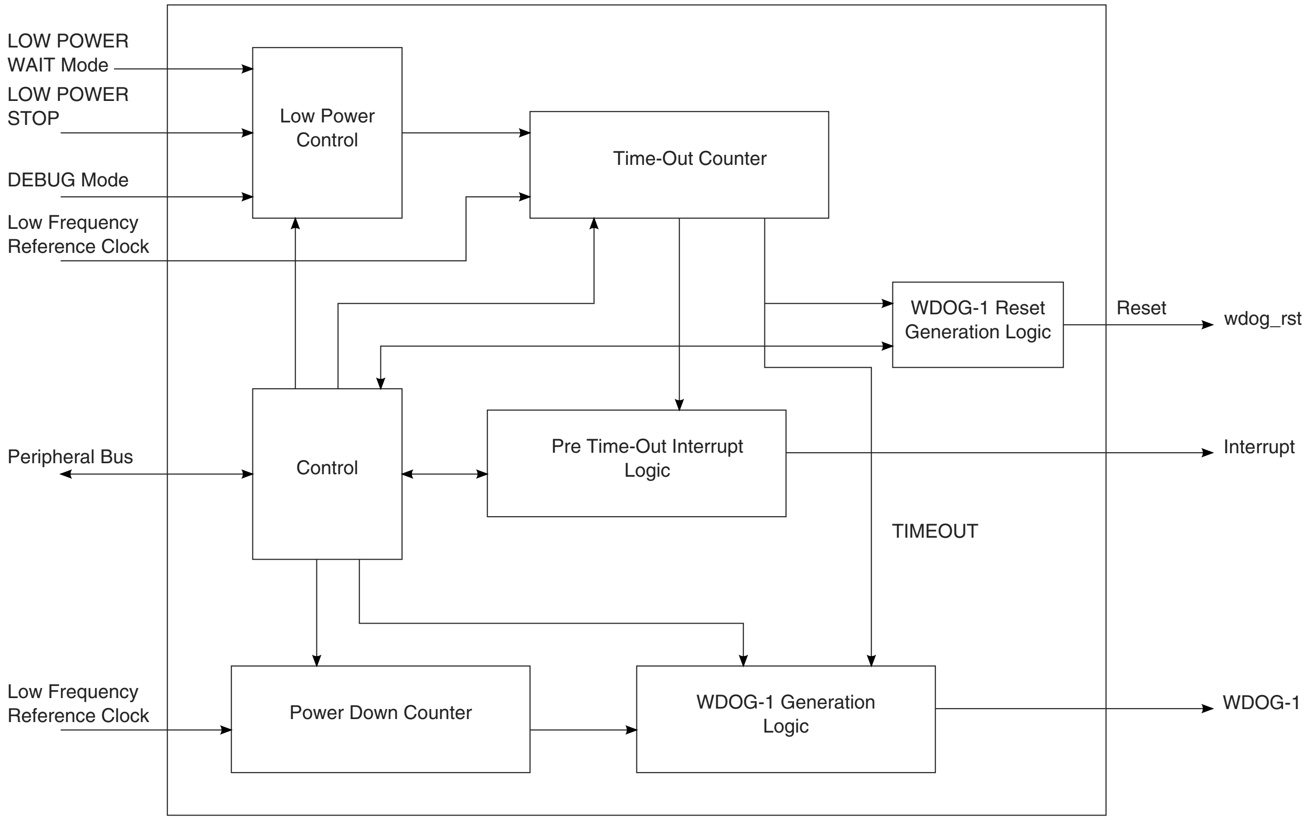


Figure 26-1.

26.1.1 Features

The WDOG features are listed below:

- Configurable timeout counter with timeout periods from 0.5 to 128 seconds which, after timeout expiration, result in the assertion of WDOG_RESET_B_DEB reset signal .
- Time resolution of 0.5 seconds
- Configurable timeout counter that can be programmed to run or stop during low-power modes
- Configurable timeout counter that can be programmed to run or stop during DEBUG mode

26.2 External signals

Table 26-1. Off-chip block signals

Signal	I/O	Description	Reset State[1]	Pull-Up/ Down
$\overline{\text{WDOG}}$	O	This signal will power down the chip. See WDOG_B generation.	1	-
$\overline{\text{wdog\_rst}}$	O	This signal is a reset source for the chip. See Watchdog reset generation.	1	-

1. The reset state values and pull-up/down requirements provided in this table are from the block-level perspective. To understand how the block is integrated at the SoC level, the system software developer must see discussions of the block in the appropriate SoC-level chapter(s). For example, a block signal that requires a pull-up could be integrated with a pull-up option built into the SoC. In this case, the system software developer must ensure the proper programming at the SoC level.

26.3 Clocks

This section describes clocks and special clocking requirements of the block.

The WDOG uses the low frequency reference clock for its counter and control operations. The peripheral bus clock is used for register read/write operations.

The low frequency reference clock is a free-running clock and can't be gated. The peripheral bus clock can't be gated and is selectively switched ON whenever read/write operations takes place.

26.4 Functional description

This section provides a complete functional description of the block.

26.4.1 Timeout event

The WDOG provides timeout periods from 0.5 to 128 seconds with a time resolution of 0.5 seconds.

The user can determine the timeout period by writing to the WDOG timeout field (WT[7:0]) in the Watchdog Control Register (WDOG_WCR). The WDOG must be enabled by setting the WDE bit of Watchdog Control Register (WDOG_WCR) for the timeout counter to start running. After the WDOG is enabled, the counter is activated, loads the timeout value and begins to count down from this programmed value. The timer will time out when the counter reaches zero and

However, the timeout condition can be prevented by reloading the counter with the new timeout value (WT[7:0] of WDOG_WCR) if a service routine (see Servicing WDOG to reload the counter) is performed before the counter reaches zero. If any system errors occur which prevent the software from servicing the Watchdog Service Register (WDOG_WSR), the timeout condition occurs. By performing the service routine, the WDOG reloads its counter to the timeout value indicated by bits WT[7:0] of the Watchdog Control Register (WDOG_WCR) and it restarts the countdown.

A system reset will reset the counter and place it in the idle state at any time during the countdown. The counter flow diagram is shown in .

NOTE

The timeout value is reloaded to the counter either at the time WDOG is enabled or after the service routine has been performed.

26.4.1.1 Servicing WDOG to reload the counter

To reload a timeout value to the counter the proper service sequence begins by writing 0x-5555 followed by 0x-AAAA to the Watchdog Service Register (WDOG_WSR). Any number of instructions can be executed between the two writes. If the WDOG_WSR is not loaded with 0x-5555 prior to writing 0x-AAAA to the WDOG_WSR, the counter is not reloaded. If any value other than 0x-AAAA is written to the WDOG_WSR after 0x-5555, the counter is not reloaded. This service sequence will reload the counter with the timeout value WT[7:0] of Watchdog Control Register (WDOG_WCR). The timeout value can be changed at any point; it is reloaded when WDOG is serviced by the core.

26.4.2 Low power modes

26.4.2.1 STOP mode

If the WDOG timer disable bit for low power STOP mode (WDZST) bit in the Watchdog Control Register (WDOG_WCR), is cleared, the WDOG timer continues to operate using the low frequency reference clock. If the low power enable (WDZST) bit is set, the WDOG timer operation will be suspended in low power STOP mode. Upon exiting low power STOP mode, the WDOG operation returns to what it was prior to entering the STOP mode.

26.4.3 Debug mode

The WDOG timer can be configured for continual operation, or for suspension during debug mode. If the WDOG debug enable (WDBG) bit is set in the Watchdog Control Register (WDOG_WCR), the WDOG timer operation is suspended in debug mode. Register read and write accesses in debug mode continue to function normally. Also, while in debug mode, the WDE bit of Watchdog Control Register (WDOG_WCR) can be enabled/disabled directly. If the WDOG debug enable (WDBG) bit is cleared then WDOG timer operation is not suspended. The power-down counter is not affected by debug mode entry/exit.

NOTE

If the WDE bit of Watchdog Control Register (WDOG_WCR) is set/cleared while in debug mode, it remains set/cleared even after exiting debug mode.

26.4.4 Operations

26.4.4.1 Watchdog reset generation

The WDOG generated reset signal $\overline{\text{wdog\_rst}}$ is asserted by the following operations:

- A software write to the Software Reset Signal (SRS) bit of the Watchdog Control Register (WDOG_WCR).
- WDOG timeout. See Timeout event.

The $\overline{\text{wdog\_rst}}$ will be asserted for one clock cycle of low frequency reference clock for a timeout condition. It remains asserted for 1 clock cycle of low frequency reference clock even if a system reset is asserted in between.

26.4.4.2 WDOG_B generation

The WDOG asserts WDOG_B in the following scenarios:

- Software write to WDA bit of Watchdog Control Register (WDOG_WCR). WDOG_B signal remains asserted as long as the WDA bit is "0".
-

26.4.5 Reset

The block is reset by a system reset and the WDOG counter will be disabled.

26.4.6 Flow Diagrams

A flow diagram of WDOG operation is shown in .

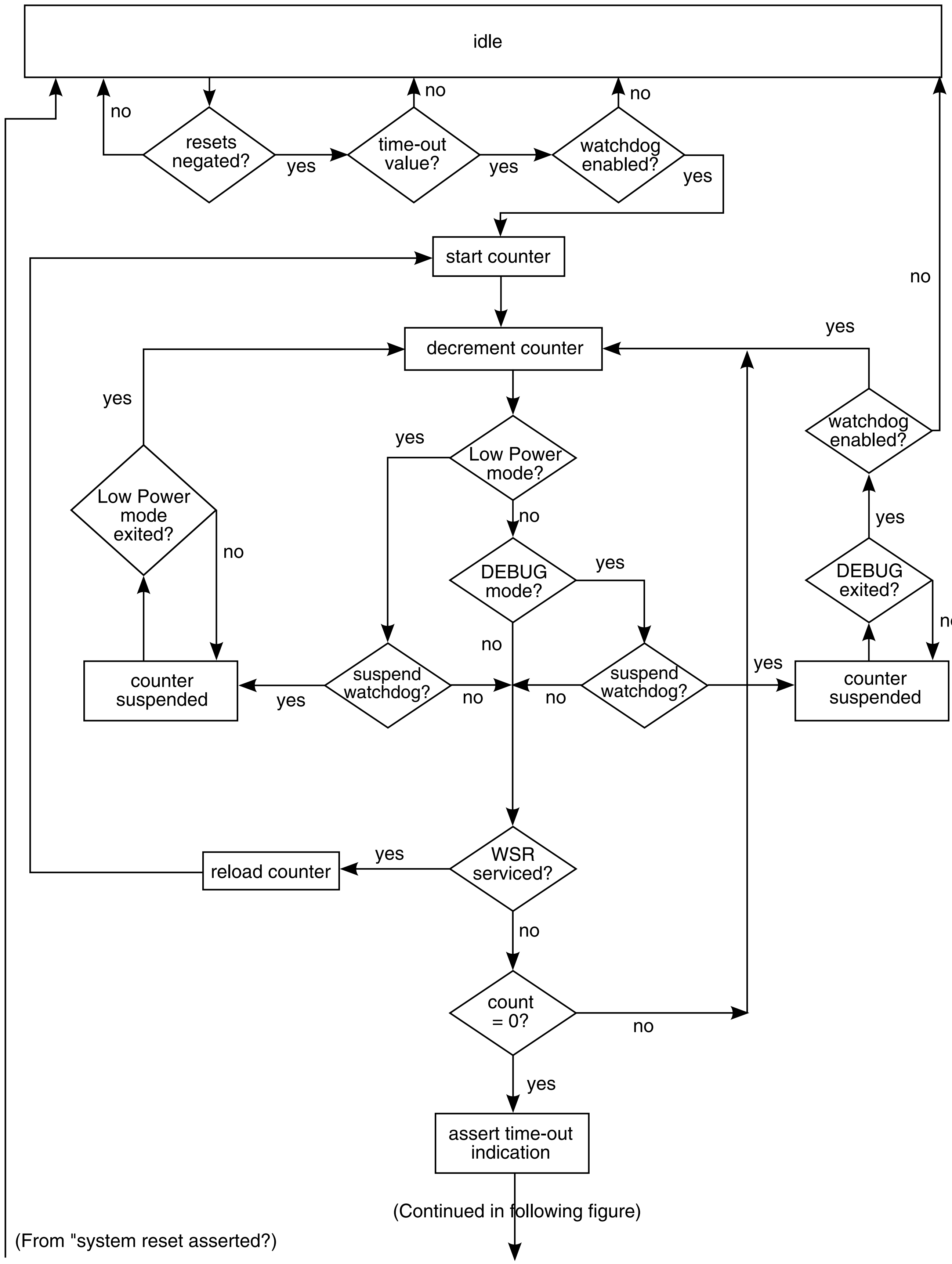


Figure 26-2. Time-Out Counter Flow Diagram

26.5 Initialization

The following sequence should be performed for WDOG initialization.

- WDT field of Watchdog Control Register (WDOG_WCR) should be programmed for sufficient timeout value.
- WDOG should be enabled by setting WDE bit of Watchdog Control Register (WDOG_WCR) so that the timeout counter loads the WDT field value of Watchdog Control Register (WDOG_WCR) and starts counting.

26.6 WDOG Memory Map/Register Definition

The WDOG has user-accessible, 16-bit registers used to configure, operate, and monitor the state of the Watchdog Timer. Byte operations can be performed on these registers. If a 32-bit access is performed,the WDOG will not generate a peripheral bus error but will behave normally, like a 16-Bit access, making read/write possible. A 32-Bit access should be avoided, as the system may go to an unknown state.

WDOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Watchdog Control Register (WDOG_WCR)	16	R/W	0030h	26.6.1/1136
2	Watchdog Service Register (WDOG_WSR)	16	R/W	0000h	26.6.2/1138
4	Watchdog Reset Status Register (WDOG_WRSR)	16	R	0000h	26.6.3/1138

26.6.1 Watchdog Control Register (WDOG_WCR)

The Watchdog Control Register (WDOG_WCR) controls the WDOG operation.

- WDE is a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared until the next system reset.

Address: 0h base + 0h offset = 0h

Bit	15	14	13	12	11	10	9	8
Read Write	WT							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	SRE	WDA	SRS	WDE	WDBG	WDZST		
Reset		0	1	1		0	0	0

WDOG_WCR field descriptions

Field	Description
15–8 WT	Watchdog Time-out Field. This 8-bit field contains the time-out value that is loaded into the Watchdog counter after the service routine has been performed or after the Watchdog is enabled. After reset, WT[7:0] must have a value written to it before enabling the Watchdog otherwise count value of zero which is 0.5 seconds is loaded into the counter. **NOTE:** The time-out value can be written at any point of time but it is loaded to the counter at the time when WDOG is enabled or after the service routine has been performed. For more information see Timeout event . 0x00 - 0.5 Seconds (Default). 0x01 - 1.0 Seconds. 0x02 - 1.5 Seconds. 0x03 - 2.0 Seconds. 0xff - 128 Seconds.
6 SRE	software reset extension, an option way to generate software reset adopt a new way to generate a more robust software reset. This bit can be set/clear with IP bus and will be reset with power-on reset . 0 using original way to generate software reset (default) 1 using new way to generate software reset.
5 WDA	WDOG_B assertion. Controls the software assertion of the WDOG_B signal. 0 Assert WDOG_B output. 1 No effect on system (Default).
4 SRS	Software Reset Signal. Controls the software assertion of the WDOG-generated reset signal $\overline{\text{wdog\_rst}}$. This bit automatically resets to "1" after it has been asserted to "0". **NOTE:** This bit does not generate the software reset to the block. 0 Assert system reset signal. 1 No effect on the system (Default).
2 WDE	Watchdog Enable. Enables or disables the WDOG block. This is a write one once only bit. It is not possible to clear this bit by a software write, once the bit is set. **NOTE:** This bit can be set/reset in debug mode (exception). 0 Disable the Watchdog (Default). 1 Enable the Watchdog.
1 WDBG	Watchdog DEBUG Enable. Determines the operation of the WDOG during DEBUG mode. This bit is write once only. 0 Continue WDOG timer operation (Default). 1 Suspend the watchdog timer.
0 WDZST	Watchdog Low Power. Determines the operation of the WDOG during low-power modes. This bit is write once-only.

Table continues on the next page...

WDOG_WCR field descriptions (continued)

Field	Description
	NOTE: The WDOG can continue/suspend the timer operation in the low-power modes (STOP mode). 0 Continue timer operation (Default). 1 Suspend the watchdog timer.

26.6.2 Watchdog Service Register (WDOG_WSR)

When enabled, the WDOG requires that a service sequence be written to the Watchdog Service Register (WSR) to prevent the timeout condition.

NOTE

Executing the service sequence will reload the WDOG timeout counter.

Address: 0h base + 2h offset = 2h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WSR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_WSR field descriptions

Field	Description
15–0 WSR	Watchdog Service Register. This 16-bit field contains the Watchdog service sequence. Both writes must occur in the order listed prior to the time-out, but any number of instructions can be executed between the two writes. The service sequence must be performed as follows: 0x5555 Write to the Watchdog Service Register (WDOG_WSR). 0xAAAA Write to the Watchdog Service Register (WDOG_WSR).

26.6.3 Watchdog Reset Status Register (WDOG_WRSR)

The WRSR is a read-only register that records the source of the output reset assertion. It is not cleared by a hard reset. Therefore, only one bit in the WRSR will always be asserted high. The register will always indicate the source of the last reset generated due to WDOG. Any write performed on this register will generate a Peripheral Bus Error .

A reset can be generated by the following sources, as listed in priority from highest to lowest:

- Watchdog Time-out
- Software Reset

Address: 0h base + 4h offset = 4h

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0			POR	0		TOUT	SFTW
Write								
Reset	0	0	0	0	0	0	0	0

WDOG_WRSR field descriptions

Field	Description
15–5 Reserved	This read-only field is reserved and always has the value 0.
4 POR	Power On Reset. Indicates whether the reset is the result of a power on reset. 0 Reset is not the result of a power on reset. 1 Reset is the result of a power on reset.
3–2 Reserved	This read-only field is reserved and always has the value 0.
1 TOUT	Timeout. Indicates whether the reset is the result of a WDOG timeout. 0 Reset is not the result of a WDOG timeout. 1 Reset is the result of a WDOG timeout.
0 SFTW	Software Reset. Indicates whether the reset is the result of a WDOG software reset by asserting SRS bit 0 Reset is not the result of a software reset. 1 Reset is the result of a software reset.

Chapter 27
Watchdog Configuration

27.1 Watchdog Scheme

The following three Watchdogs are included in this device:

a) General Purpose WDOG (WDOG-A5) - Monitors A5

b) General Purpose WDOG (WDOG-M4) - Monitors CM4

c) TrustZone Watchdog (WDOG - SNVS) - Part of TrustZone and monitors Secure world

NOTE

Above does not include another WDOG that is part of A5 core complex (CA5-internal) that runs on core clock.

WDOG_SNVS is another instance of the general purpose WDOG mainly intended for secure platforms and with the WDOG reset being controlled by CSU. Since WDOG_SNVS monitors secure word, TZ-WDOG does not have any significance for non-secure parts.

Table 27-1. WDOG Configuration

Configuration	Security	WDOG-SNVS	WDOG-A5	WDOG-M4	Comments
A5 only configuration	Enabled	Functional	Functional	Disabled	
A5 only configuration	Disabled	Disabled	Functional	Disabled	
M4 only Configuration	Enabled	Enabled	Disabled	Functional	
M4 only Configuration	Disabled	Disabled	Disabled	Functional	
Dual Core(A5 Primary)	Enabled	Functional	Functional	Functional	

Table continues on the next page...

Table 27-1. WDOG Configuration (continued)

Configuration	Security	WDOG-SNVS	WDOG-A5	WDOG-M4	Comments
Dual Core (A5 Primary)	Disabled	Disabled	Functional	Functional	
Dual Core (M4 Primary)	Enabled	INVALID Configuration			Security cannot be enabled with M4 as primary core
Dual Core (M4 Primary)	Disabled	Disabled	Functional	Functional	Security Switch should automatically disable or ignore WDOG_SNVS

NOTE

1. TZ-WDOG is only relevant for secure parts as this part of Trust Zone.

All WDOGs are disabled by default (after the reset). WDOG-A5, WDOG-M4 will be configured during the boot while WDOG-SNVS is dedicated for secure world purposes and will be activated by TZ Software if required.

- If servicing does not take place, the timer times out and WDOG reset signal activated
- Interrupt can be generated before the counter actually times out
- wdog_rst_b signal can be activated by Software
- There is a power down counter which gets enabled out of any reset. This counter has a fixed time out period of 16 seconds upon which it will assert the ipp_wdog_b signal

27.2 Watchdog Connectivity

The following Figure shows the way different WDOG on this device needs to be connected to GIC, NVIC, SRC , CSU and the pins.

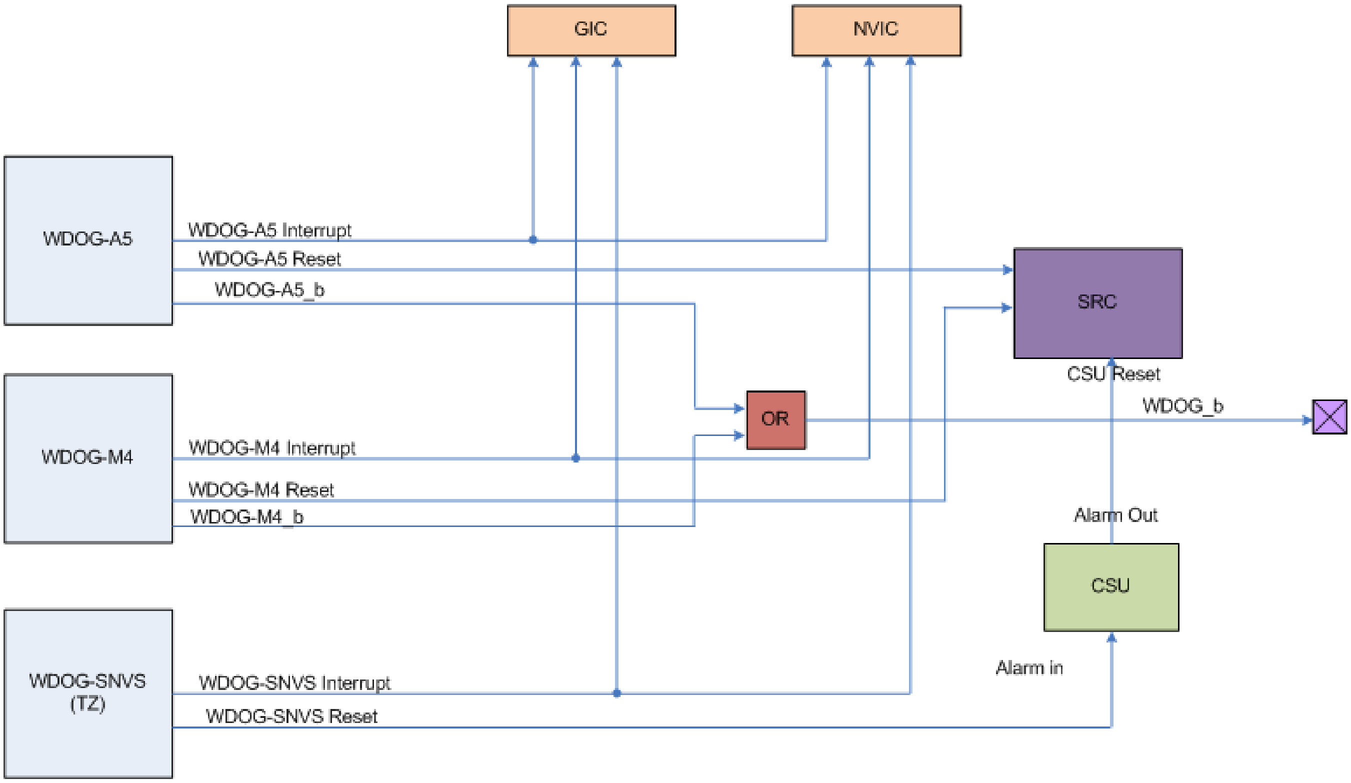


Figure 27-1. WDOG Connectivity

Any WDOG that times out (except WDOG_SNVS) must generate a reset to Reset controller(SRC). This is applicable for both conditions a) where A5 is primary and M4 is secondary b) where M4 is Primary and A5 is secondary.

NOTE

This device does not support a system recovery where A5 only domain or M4 only domain is reset. Either WDOG-A5 or WDOG-M4 generated reset should eventually RESET the system.

Reset generated by the WDOG_SNVS is moderated by CSU.

WDOG_b output to the pins could be optionally used by extenal power management IC to turn off/control the power to the chip.

27.3 WDOG clocking options

WDOG-A5 WDOG-M4 & WDOG-SNVS should supply following clock options

a) 128K IRC

b) 32K Crystal

c) Bus clock (Divided)

Default option MUST be 128 IRC.

27.4 WDOG Debug requirement

1. The CM4 watchdog(WDOG-M4) should freeze if the CM4 enters debug halt mode.
2. The CA5 watchdog(WDOG-A5) should freeze if the CA5 enters debug halt mode.

The above requirements are on top on what is already supported by the IP.

Chapter 28
Wakeup Unit (WKPU)

28.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The WKPU supports 16 external sources that can generate interrupts or wakeup events and 1 external source(s) that can cause non-maskable interrupt request(s) or wakeup event(s). In addition, it combines its wakeup events with those generated from 5 other wakeup sources to supply a single wakeup to the system. The block diagram of the WKPU and its interfaces to other system components is shown below.

NOTE

The signal widths in the following diagram might not depict this device's particular configuration. See the chip configuration chapter for more information.

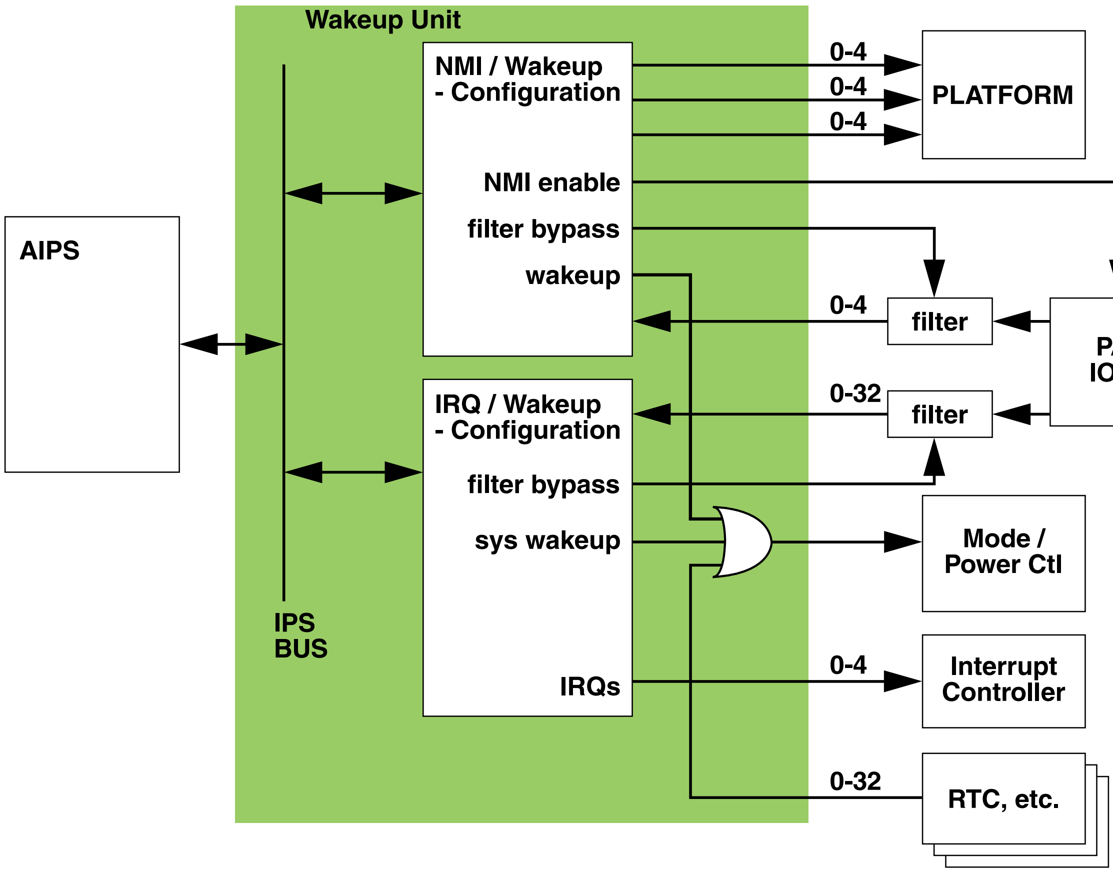


Figure 28-1. WKPU and connected system components

28.2 Features

The WKPU supports these features:

- Non-maskable Interrupt support with:
    - One external NMI source
    - One analog glitch filter
- Independent interrupt destination for each core:
    - Non-maskable interrupt
- Active edge selection control for events to each core
- Configurable system wakeup triggering from NMI source(s)
- External wakeup/interrupt support with:
    - One system interrupt vector for interrupt sources

  - 16 analog glitch filters
  - Independent interrupt mask
  - Edge detection
  - Configurable system wakeup triggering from all interrupt sources
  - Configurable pullup
- On-chip wakeup support:
  - Five wakeup sources
  - Wakeup status mapped to same register as external wakeup/interrupt status

28.3 External signal description

The WKPU has 16 signal inputs that can be used as external interrupt sources in normal run mode or as system wakeup sources in certain power down modes.

The module has 1 signal input(s) that can be used as non-maskable interrupt source(s) in normal run mode or as system wakeup sources in certain power down modes.

28.4 WKPU memory map and registers

This section provides a detailed description of all registers accessible in the WKPU module.

NOTE

Reserved registers will read as 0, writes will have no effect. If supported and enabled by the SoC, a transfer error will be issued when trying to access completely reserved register space.

WKPU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A000	NMI Status Flag Register (WKPU_NSR)	32	R/W	0000_0000h	28.4.1/1148
4006_A008	NMI Configuration Register (WKPU_NCR)	32	R/W	0000_0000h	28.4.2/1149
4006_A014	Wakeup/Interrupt Status Flag Register (WKPU_WISR)	32	R/W	0000_0000h	28.4.3/1150

Table continues on the next page...

WKPU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A018	Interrupt Request Enable Register (WKPU_IRER)	32	R/W	0000_0000h	28.4.4/1151
4006_A01C	Wakeup Request Enable Register (WKPU_WRER)	32	R/W	0000_0000h	28.4.5/1152
4006_A028	Wakeup/Interrupt Rising-Edge Event Enable Register (WKPU_WIREER)	32	R/W	0000_0000h	28.4.6/1152
4006_A02C	Wakeup/Interrupt Falling-Edge Event Enable Register (WKPU_WIFEER)	32	R/W	0000_0000h	28.4.7/1153
4006_A030	Wakeup/Interrupt Filter Enable Register (WKPU_WIFER)	32	R/W	0000_0000h	28.4.8/1153
4006_A034	Wakeup/Interrupt Pullup Enable Register (WKPU_WIPUER)	32	R/W	0000_0000h	28.4.9/1154

28.4.1 NMI Status Flag Register (WKPU_NSR)

This register holds the non-maskable interrupt status flags.

NOTE

This register is accessible by 8-, 16-, and 32-bit read/write operations

Address: 4006_A000h base + 0h offset = 4006_A000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NIF0	NOVF0	0						0		0					
W	w1c	w1c														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0						0		0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKPU_NSR field descriptions

Field	Description
31 NIF0	NMI Status Flag 0. This flag can be cleared only by writing a 1. Writing a 0 has no effect. If enabled (NREE0 or NFEE0 set), NIF0 causes an interrupt request.

Table continues on the next page...

WKPU_NSR field descriptions (continued)

Field	Description
	0 No event has occurred on the pad 1 An event as defined by NREE0 and NFEE0 has occurred
30 NOVF0	NMI Overrun Status Flag 0. This flag can be cleared only by writing a 1. Writing a 0 has no effect. It will be a copy of the current NIF0 value whenever a NMI event occurs, thereby indicating to the software that a NMI occurred while the last one was not yet serviced. If enabled (NREE0 or NFEE0 set), NOVF0 causes an interrupt request. 0 No overrun has occurred on NMI input 0 1 An overrun has occurred on NMI input 0
29–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

28.4.2 NMI Configuration Register (WKPU_NCR)

This register holds the configuration bits for the non-maskable interrupt settings.

NOTE

This register is accessible by 8-, 16-, and 32-bit read/write operations

NOTE

Writing a '0' to both NREE[x] and NFEE[x] disables the NMI functionality completely (i.e. no system wakeup or interrupt will be generated on any pad activity)

Address: 4006_A000h base + 8h offset = 4006_A008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NLOCK0	NDSS0		NWRE0	0	NREE0	NFEE0	NFE0	0							
W																
Reset	0			0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKPU_NCR field descriptions

Field	Description
31 NLOCK0	NMI Configuration Lock Register 0. Writing a 1 to this bit locks the configuration for the NMI until it is unlocked by a system reset or STANDBY0 mode exit. Writing a 0 has no effect.
30–29 NDSS0	NMI Destination Source Select 0. This value is reset on STANDBY0 mode entry . 00 non-maskable interrupt 01 Reserved 10 Reserved 11 Reserved
28 NWRE0	NMI Wakeup Request Enable 0. 0 System wakeup requests from the corresponding NIF0 bit are disabled 1 A set NIF0 bit or set NOVF0 bit causes a system wakeup request
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 NREE0	NMI Rising-edge Events Enable 0. 0 Rising-edge event is disabled 1 Rising-edge event is enabled
25 NFEE0	NMI Falling-edge Events Enable 0. 0 Falling-edge event is disabled 1 Falling-edge event is enabled
24 NFE0	NMI Filter Enable 0. Enable analog glitch filter on the NMI pad input. 0 Filter is disabled 1 Filter is enabled
23–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

28.4.3 Wakeup/Interrupt Status Flag Register (WKPU_WISR)

This register holds the wakeup/interrupt flags.

NOTE

- This register is accessible only by 32-bit read/write operations
- Status bits associated with on-chip wakeup sources are located to the left of the external wakeup/interrupt status bits and are read only. The wakeup for these sources must be configured and cleared at the on-chip wakeup source. Also, the configuration registers for the external interrupts/wakeups do not have corresponding bits.

Address: 4006_A000h base + 14h offset = 4006_A014h

Bit	31–0
R	EIF
W	w1c
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

WKPU_WISR field descriptions

Field	Description
31–0 EIF	External Wakeup/Interrupt Status Flag x. This flag can be cleared only by writing a 1. Writing a 0 has no effect. If enabled (IRER[x]), EIF[x] causes an interrupt request. 0 No event has occurred on the pad 1 An event as defined by WIREER and WIFEER has occurred

28.4.4 Interrupt Request Enable Register (WKPU_IRER)

This register is used to enable the interrupt messaging from the wakeup/interrupt pads to the interrupt controller.

NOTE

This register is accessible only by 32-bit read/write operations

Address: 4006_A000h base + 18h offset = 4006_A018h

Bit	31–0
R W	EIRE
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

WKPU_IRER field descriptions

Field	Description
31–0 EIRE	External Interrupt Request Enable x. 0 Interrupt requests from the corresponding EIF[x] bit are disabled 1 A set EIF[x] bit causes an interrupt request

28.4.5 Wakeup Request Enable Register (WKPU_WRER)

This register is used to enable the system wakeup messaging from the wakeup/interrupt pads to the mode entry and power control modules.

NOTE

This register is accessible only by 32-bit read/write operations

If a pin is disabled through this register, the corresponding bits in the WIFEER and WIREER registers must be written to 0 to ensure that the pin does not respond to any change.

Address: 4006_A000h base + 1Ch offset = 4006_A01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WRE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKPU_WRER field descriptions

Field	Description
31–0 WRE	External Wakeup Request Enable x. 0 System wakeup requests from the corresponding EIF[x] bit are disabled 1 A set EIF[x] bit causes a system wakeup request

28.4.6 Wakeup/Interrupt Rising-Edge Event Enable Register (WKPU_WIREER)

This register is used to enable rising-edge triggered events on the corresponding wakeup/ interrupt pads.

NOTE

This register is accessible only by 32-bit read/write operations

Address: 4006_A000h base + 28h offset = 4006_A028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IREE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKPU_WIREER field descriptions

Field	Description
31–0 IREE	External Interrupt Rising-edge Events Enable x. 0 Rising-edge event is disabled 1 Rising-edge event is enabled

28.4.7 Wakeup/Interrupt Falling-Edge Event Enable Register (WKPU_WIFEER)

This register is used to enable falling-edge triggered events on the corresponding wakeup/ interrupt pads.

NOTE

This register is accessible only by 32-bit read/write operations

Address: 4006_A000h base + 2Ch offset = 4006_A02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IFEEx																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKPU_WIFEER field descriptions

Field	Description
31–0 IFEEx	External Interrupt Falling-edge Events Enable x. 1 Falling-edge event is enabled 0 Falling-edge event is disabled

28.4.8 Wakeup/Interrupt Filter Enable Register (WKPU_WIFER)

This register is used to enable an analog filter on the corresponding interrupt pads to filter out glitches on the inputs. The number of wakeups/interrupts supporting this feature is SoC dependent and can be configured to be between 1 and 32.

NOTE

This register is accessible only by 32-bit read/write operations

Address: 4006_A000h base + 30h offset = 4006_A030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IFE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKPU_WIFER field descriptions

Field	Description
31–0 IFE	External Interrupt Filter Enable x. Enable analog glitch filter on the external interrupt pad input. 0 Filter is disabled 1 Filter is enabled

28.4.9 Wakeup/Interrupt Pullup Enable Register (WKPU_WIPUER)

This register is used to enable a pullup on the corresponding interrupt pads to pull an unconnected wakeup/interrupt input to a value of '1'.

NOTE

This register is accessible only by 32-bit read/write operations

Address: 4006_A000h base + 34h offset = 4006_A034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IPUE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WKPU_WIPUER field descriptions

Field	Description
31–0 IPUE	External Inpterrupt Pullup Enable x. 0 Pullup is disabled 1 Pullup is enabled

28.5 Functional description

This section provides a functional description of the WKPU.

28.5.1 Non-maskable interrupts

The WKPU supports the capturing of a second event per NMI input before the interrupt is cleared, thus reducing the chance of losing an NMI event.

Each NMI passes through a bypassable analog glitch filter.

NOTE

Glitch filter control and pad configuration should be done while the NMI is disabled in order to avoid erroneous triggering by glitches caused by the configuration process itself.

NOTE

The figure below represents a generic configuration and might not represent this particular device's configuration. See the chip-specific information for details on this chip's WKPU.

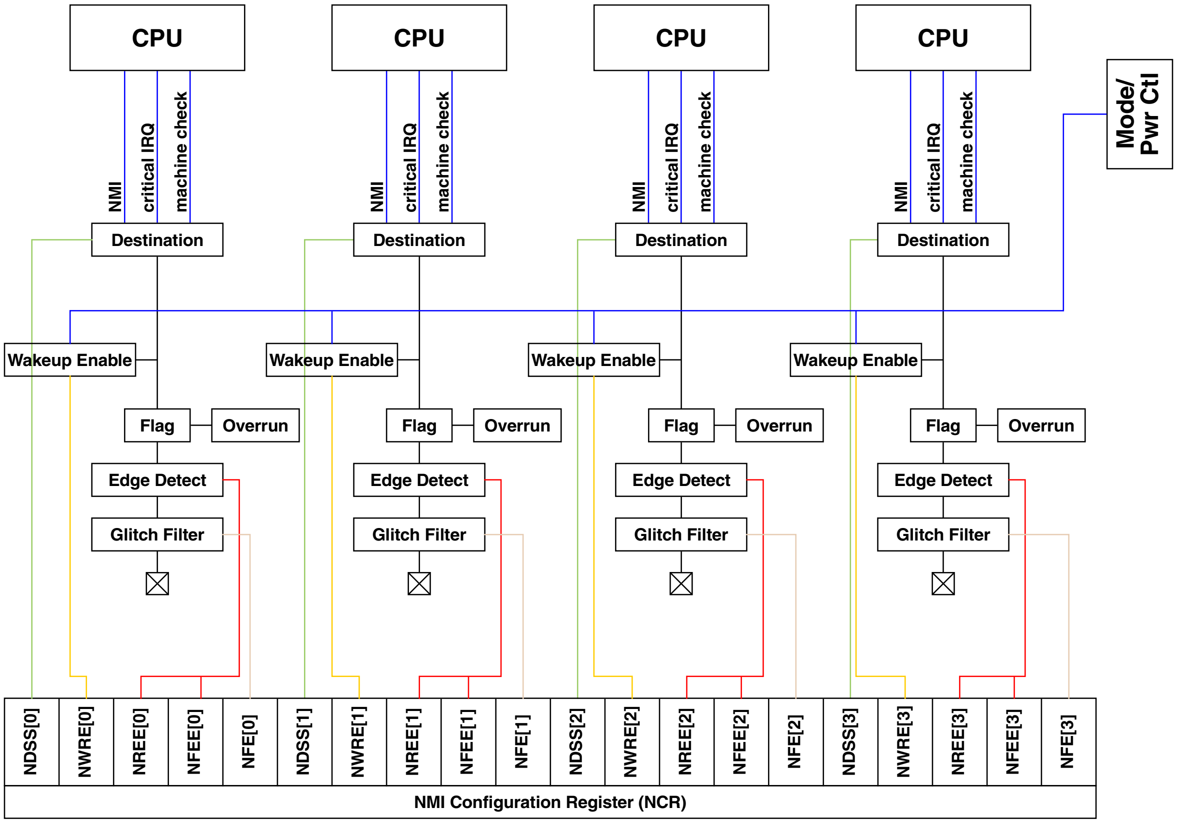


Figure 28-11. NMI pad diagram

28.5.1.1 NMI management

Each NMI can be enabled or disabled independently. This can be performed using the single NCR register laid out to contain all configuration bits for a given NMI in a single byte (see NMI Configuration Register (WKPU_NCR)). A pad defined as an NMI can be configured by the user to recognize interrupts with an active rising edge, an active falling edge or both edges being active. A setting of having both edge events disabled results in no interrupt being detected and should not be configured.

The active NMI edge is controlled by the user through the configuration of the NREE and NFEE bits.

Note

> After reset, NREE and NFEE are set to '0', therefore the NMI functionality is disabled after reset and must be enabled explicitly by software.

Once a pad's NMI functionality has been enabled, the pad cannot be reconfigured in the IOMUX to override or disable the NMI.

The NMI destination interrupt is controlled by the user through the configuration of the NDSS bits. See NMI Configuration Register (WKPU_NCR) for details.

Each NMI supports a status flag and an overrun flag which are located in the NSR register (see NMI Status Flag Register (WKPU_NSR)). This register is a clear-by-write-1 register type, preventing inadvertent overwriting of other flags in the same register. The status flag is set whenever an NMI event is detected. The overrun flag is set whenever an NMI event is detected and the status flag is set (i.e. has not yet been cleared).

Note

> The overrun flag is cleared by writing a '1' to the appropriate overrun bit in the NSR register. If the status bit is cleared and the overrun bit is still set, the pending interrupt will not be cleared.
>
> During an NMI ISR, on wakeup of the SoC from an NMI, any writes to ECC-protected memory must have the correct ECC.

28.5.2 External wakeups/interrupts

The WKPU supports 1 interrupt vector(s) to the interrupt controller of the SoC. Each interrupt vector can support up to the number of external interrupt sources from the device pads with the total across all vectors being equal to the number of external interrupt sources. Each external interrupt sources is assigned to exactly one interrupt vector. The interrupt vector assignment is sequential so that one interrupt vector is for external interrupt sources 0 through N-1, the next is for N through N+M-1, and so forth.

Refer to the following figure for an overview of the external interrupt implementation for the example of four interrupt vectors with eight external interrupt sources each.

NOTE

The figure below shows a generic representation of the WKPU. See the chip configuration chapter for details on how this device's configuration might differ from this figure.

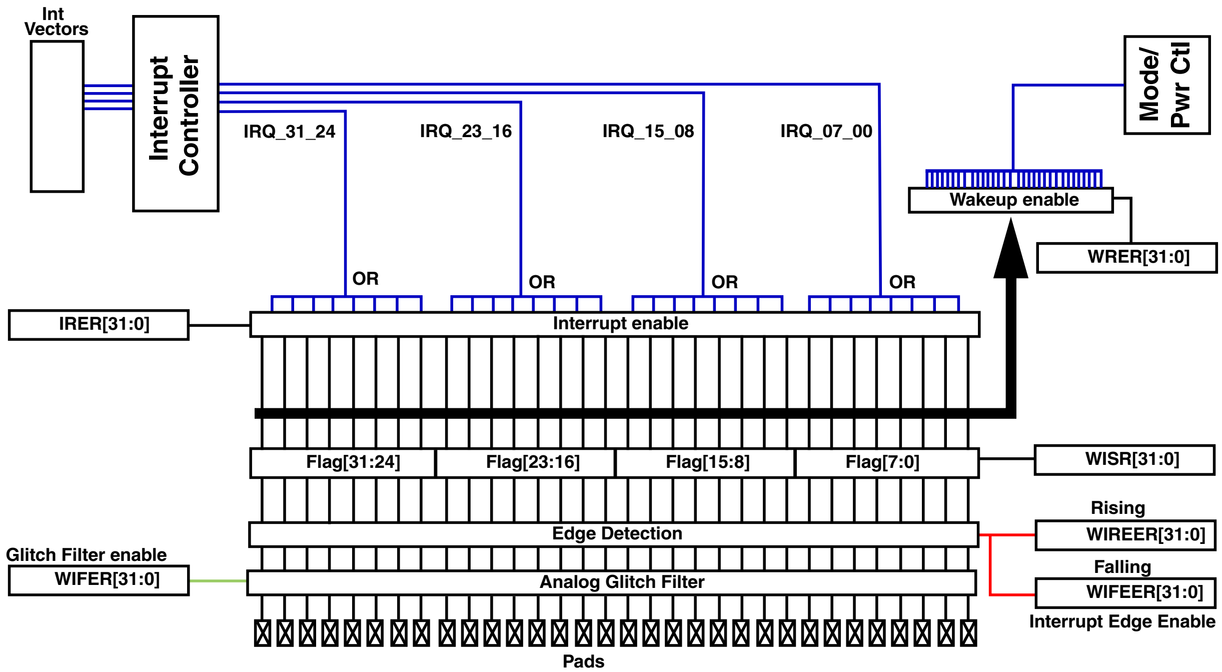


Figure 28-12. External interrupt pad diagram

All of the external interrupt pads within a single group have equal priority. It is the responsibility of the user software to search through the group of sources in the most appropriate way for their application.

The priority of the vectors used by the external interrupt pads is fixed based on the platform and the interrupt controller and its priority levels, but the allocation of pads to each group of interrupts can be independently configured by the SoC.

External interrupt lines have a digital glitch filters applied to them.

28.5.2.1 External interrupt management

Each interrupt can be enabled or disabled independently. This can be performed using a single rolled up register (Interrupt Request Enable Register (WKPU_IRER)). A pad defined as an external interrupt can be configured by the user to recognize interrupts with an active rising edge, an active falling edge or both edges being active.

Note

> Writing a '0' to both IREE[x] and IFEE[x] disables the external interrupt functionality for that pad completely (i.e. no system wakeup or interrupt will be generated on any activity on that pad)!

The active IRQ edge is controlled by the users through the configuration of the registers WIREER and WIFEER.

Each external interrupt supports an individual flag which is held in the flag register (WISR). This register is a clear-by-write-1 register type, preventing inadvertent overwriting of other flags in the same register.

28.5.3 On-chip wakeups

The WKPU supports 5 on-chip wakeup sources. It combines the on-chip wakeups with the external ones to generate a single wakeup to the system.

28.5.3.1 On-chip wakeup management

In order to allow software to determine the wakeup source at one location, on-chip wakeups are reported along with external wakeups in the WISR register (see Wakeup/ Interrupt Status Flag Register (WKPU_WISR) for details). Enabling and clearing of these wakeups are done via the on-chip wakeup source's own registers.

28.6 Initialization Information

This section discusses initialization for the following features:

- Glitch Filter and Pad Configuration
- Non-maskable interrupt initialization
- Reset Request

28.6.1 Glitch Filter and Pad Configuration

Glitch filter control and pad configuration should be performed while the NMI is disabled in order to avoid erroneous triggering by glitches caused by the configuration process itself.

When enabling the glitch filter, do not enable the rising-/falling-edge events bits, i.e., the NREE, NFEE, RREE, and RFEE, bits in the same register write.

28.6.2 Non-Maskable Interrupts

When an NMI interrupt pin is first enabled, it is possible to get a false interrupt flag. To prevent a false interrupt request during pin interrupt initialization the user must do the following:

1. Mask interrupts by clearing NSR[NIF*n*] and NSR[NOVF*n*].
2. Clear NCR[NFEE*n*] and NCR[NREE*n*], preferably by writing 0 to all bits in the register.
3. Configure the analog glitch filter using NCR[NFE0].
4. Configure the appropriate SIUL2_MSCR register for the desired NMI interrupt pin(s) as follows:
    - Clear the ODC bits to disable output.
    - Set the IBE bit to enable the pin's input buffer.
    - If the internal weak pull-up/pull-down is used, configure the appropriate WPUE and WPDE bits.

    NOTE

    NMI interrupt pins should never be configured as outputs, i.e., MSCR[ODC] bits are not zeros, when external interrupt inputs are desired since false interrupts could be detected (such as from a GPIO configuration).

5. Write appropriate bit values to the NCR register to configure:
    - NMI source (NDSS*n*).
    - Wakeup request (NWRE*n*).

- Glitch filter (NFE0)
- Rising/falling edge events enable (NFEE*n* and NREE*n*).
- NMI configuration lock (NLOCK*n*).

28.6.3 Reset Request

When an NMI reset request pin is first enabled, it is possible to get a false reset request. To prevent a false reset request during pin reset request initialization, the user must do the following:

1. Mask reset requests by clearing NSR[RIF] and NSR[ROVF].
2. Clear NCR[RFEE] and NCR[RREE], preferably by writing 0 to all bits in this register.
3. Configure the analog glitch filter, as desired, using NCR[NFE0].
4. Configure the appropriate SIUL2_MSCR register for the NMI interrupt pin(s) as follows:
    - Clear the ODC bits to disable output.
    - Set the IBE bit to enable the pin's input buffer.
    - If the internal weak pull-up/pull-down is used, configure the appropriate WPUE and WPDE bits.

NOTE

NMI reset request pins should never be configured as outputs (i.e., MSCR[ODC] bits are not zeros) when external reset request inputs are desired since false reset requests could be detected (e.g., from a GPIO configuration).

5. Write appropriate bit values to the NCR register to configure:
    - Reset destination source (RDSS]).
    - Reset wakeup request (RWRE).
    - Glitch filter (NFE0)

- Rising/falling edge events enable (RFEE and RREE).
- Reset configuration lock (RLOCK).

Chapter 29
Local Memory Controller

29.1 Introduction

The Local Memory Controller provides the ARM®Cortex-M4™ processor with tightly-coupled processor-local memories and bus paths to all slave memory spaces.

29.1.1 Block Diagram

The Cortex-M4 processor has a modified 32-bit Harvard bus architecture. Using a 32-bit address space, low-order addresses (0x0000_0000 through 0x1FFF_FFFF) use the Processor Code (PC) bus, and high-order addresses (0x2000_0000 through 0xFFFF_FFFF) use the Processor System (PS) bus. As the bus names imply, normal operation has code accesses on the PC bus and data accesses on the PS bus.

This device has been augmented with tightly-coupled memories for the PC and PS buses. The memories include RAMs and caches. These local memories provide zero wait state access to RAM and cacheable address spaces.

The local memory controller includes four memory controllers and their attached memories:

- SRAM lower (SRAM_L) controller via the PC bus
- SRAM upper (SRAM_U) controller via the PS bus
- Cache memory controller via the PC bus
- Cache memory controller via the PS bus

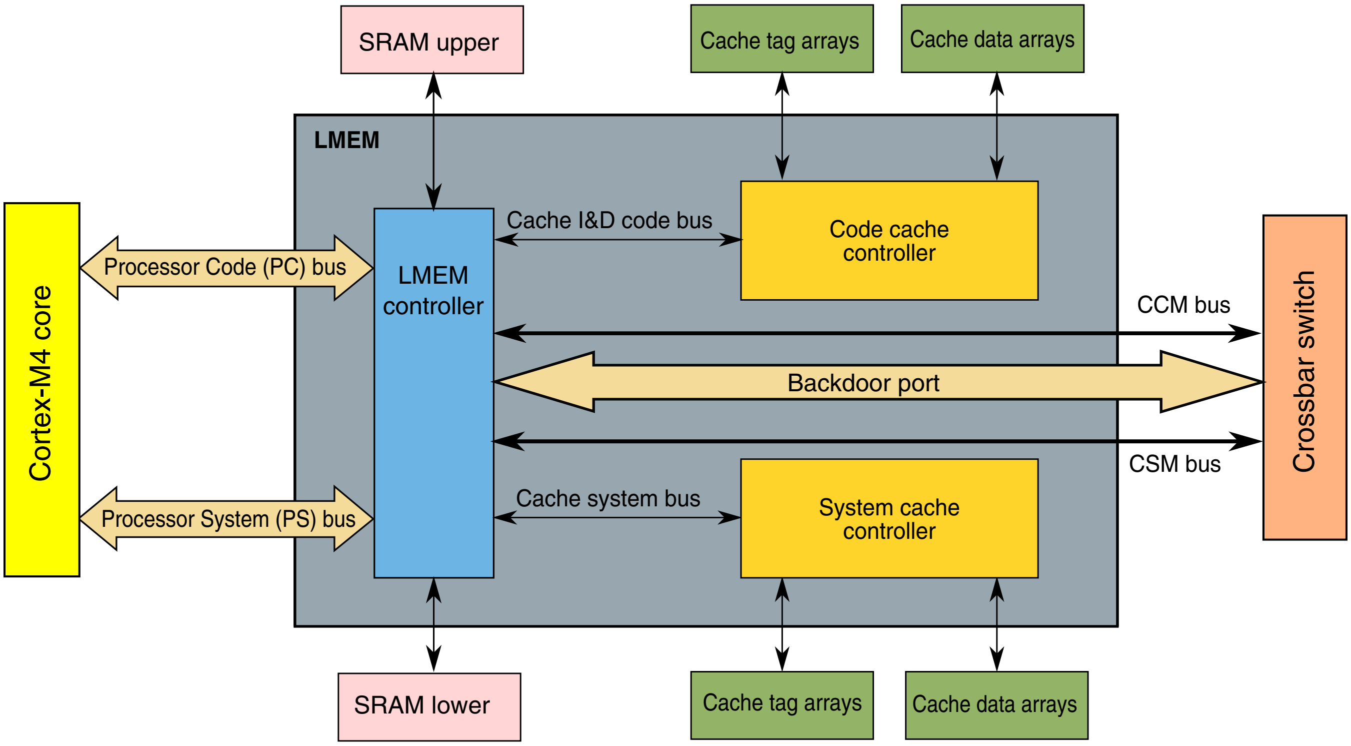


Figure 29-1. Local memory controller block diagram

NOTE

The SRAM and cache controllers reside within the LMEM, but the single-port synchronous RAM arrays used by these controllers are external.

The LMEM contains address decode logic for the PC and PS buses. This logic routes the core's accesses to the various system resources. The address spaces are device-specific and are specified in the device's Chip Configuration chapter.

The CM4 core address space is decoded as follows:

Table 29-1. Address Space Decode

Address Range	Core/Slave Bus[1]	Slave Target, XBAR Port	Cache Mode at reset
0x0000_0000 → 0x07FF_FFFF	PC/CCM	Boot Rom (S0A)	cacheable write-through
0x0800_0000 → 0x0FFF_FFFF	PC/CCM	DDR code alias (S8)	cacheable write-through
0x1000_0000 → 0x17FF_FFFF	PC/CCM	QuadSPI0 alias (S2)	cacheable write-through
0x1800_0000 → 0x1EFF_FFFF	PC/CCM	FlexBus code alias (S0B)	cacheable write-through
0x1F00_0000→0x1F7F_FFFF	PC/CCM	OCRAM code alias (S3/S4/S5)	cacheable write-through
0x1F80_0000→0x1FFF_FFFF	PC/---	CM4_TCML Frontdoor	non-cacheable
0x2000_0000 → 0x2FFF_FFFF	PS/CSM	QuadSPI0 (S2)	cacheable write-back

Table continues on the next page...

Table 29-1. Address Space Decode (continued)

Address Range	Core/Slave Bus[1]	Slave Target, XBAR Port	Cache Mode at reset
0x3000_0000 → 0x3EFF_FFFF	PS/CSM	FlexBus (S0B)	cacheable write-back
0x3F00_0000 → 0x3F7F_FFFF	PS/CSM	OCRAM (S3/S4/S5)	cacheable write-back
0x3F80_0000 → 0x3FFF_FFFF	PS/---	CM4_TCMU Frontdoor	non-cacheable
0x4000_0000 → 0x4006_FFFF	PS/CSM	IPS0 Peripherals (S0C)	non-cacheable
0x4007_0000 → 0x4007_FFFF	PS/CSM	Secure RAM (S6)	non-cacheable
0x4008_0000 → 0x400F_EFFF	PS/CSM	IPS1 Peripherals (S1C)	non-cacheable
0x400F_0000 → 0x400F_FFFF	PS/CSM	RGPIO (S1A)	non-cacheable
0x4010_0000 → 0x4FFF_FFFF	PS/CSM	NOT USED	–
0x5000_0000 → 0x5FFF_FFFF	PS/CSM	QuadSPI1 (S7A)	cacheable
0x6000_0000 → 0x77FF_FFFF	PS/CSM	NOT USED	–
0x7800_0000→0x79FF_FFFF	PS/CSM	RLE (S7B)	cacheable write-through
0x7A00_0000→0x7BFF_FFFF	PS/CSM	QS1 RxBuf (S7A)	cacheable write-through
0x7C00_0000→0x7DFF_FFFF	PS/CSM	QS0 RxBuf (S2)	cacheable write-through
0x7E00_0000→0x7FFF_FFFF	PS/CSM	altGFXRAM (S5)	cacheable write-through
0x8000_0000→0xDFFF_FFFF	PS/CSM	DDR (S8)	cacheable write-back
0xE000_0000→0xE00F_FFFF	2	CM4 PPB	–

1. All references are routed to the core's Private Peripheral Bus (PPB) and never appear on the PS/CSM buses.
2. All references are routed to the core's Private Peripheral Bus (PPB) and never appear on the PS/CSM buses.

29.1.2 Cache features

A cache is a block of high-speed memory locations containing address information (commonly known as a tag) and the associated data. The purpose is to decrease the average time of a memory access. Caches operate on two principles of locality:

- Spatial locality — An access to one location is likely to be followed by accesses from adjacent locations (for example, sequential instruction execution or usage of a data structure).
- Temporal locality — An access to an area of memory is likely to be repeated within a short time period (for example, execution of a code loop).

To minimize the quantity of control information stored, the spatial locality property is used to group several locations together under the same tag. This logical block is commonly known as a cache line.

When data is loaded into a cache, access times for subsequent loads and stores are reduced, resulting in overall performance benefits. An access to information already in a cache is known as a cache hit, and other accesses are called cache misses.

Normally, caches are self-managing, with the updates occurring automatically. Whenever the processor wants to access a cacheable location, the cache is checked. If the access is a cache hit, the access occurs immediately. Otherwise, a location is allocated and the cache line is loaded from memory. Different cache topologies and access policies are possible. However, they must comply with the memory coherency model of the underlying architecture.

Caches introduce a number of potential problems, mainly because of:

- memory accesses occurring at times other than when the programmer would normally expect them,
- the existence of multiple physical locations where a data item can be held.

The local memory controller supports three modes of operation:

1. Write-through — access to address spaces with this cache mode are cacheable.
    - A read miss on the input bus causes a line read on the output busof a 32-byte-aligned memory address containing the desired address. This miss data is loaded into the cache and is marked as valid and not modified.
    - A write-through read hit to a valid cache location returns data from the cache with no output bus access.
    - A write-through write miss bypasses the cache and writes to the output bus (no allocate on write miss policy for write-through mode spaces).
    - A write-through write hit updates the cache hit data and writes to the output bus.
2. Write-back — access to address spaces with this cache mode are cacheable.
    - A write-back read miss on the input bus will cause a line read on the output bus of a 32-byte-aligned memory address containing the desired address. This miss data is loaded into the cache and marked as valid and not modified.
    - A write-back read hit to a valid cache location will return data from the cache with no output bus access.
    - A write-back write miss will do a "read-to-write" (allocate on write miss policy for write-back mode spaces). A line read on the output bus of a 16 byte aligned memory address containing the desired write address is performed. This miss data is loaded into the cache and marked as valid and modified; and the write data will then update the appropriate cache data locations.
3. Non-cacheable — access to address spaces with this cache mode are not cacheable. These accesses bypass the cache and access the output bus.

29.2 Memory Map/Register Definition

The cache programmer's model provides a variety of registers for configuring and controlling the cache, as well as indirect access paths to all cache tag and data storage.

LMEM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_2000	Cache control register (LMEM_PCCCR)	32	R/W	0000_0000h	29.2.1/1167
E008_2004	Cache line control register (LMEM_PCCLCR)	32	R/W	0000_0000h	29.2.2/1168
E008_2008	Cache search address register (LMEM_PCCSAR)	32	R/W	0000_0000h	29.2.3/1171
E008_200C	Cache read/write value register (LMEM_PCCCVR)	32	R/W	0000_0000h	29.2.4/1172
E008_2800	Cache control register (LMEM_PSCCR)	32	R/W	0000_0000h	29.2.5/1172
E008_2804	Cache line control register (LMEM_PSCLCR)	32	R/W	0000_0000h	29.2.6/1174
E008_2808	Cache search address register (LMEM_PSCSAR)	32	R/W	0000_0000h	29.2.7/1176
E008_280C	Cache read/write value register (LMEM_PSCCVR)	32	R/W	0000_0000h	29.2.8/1177

29.2.1 Cache control register (LMEM_PCCCR)

Address: E008_2000h base + 0h offset = E008_2000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GO	0			PUSHW1	INVW1	PUSHW0	INVW0	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												PCCR3	PCCR2	ENWRBUF	ENCACHE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PCCCR field descriptions

Field	Description
31 GO	Initiate Cache Command Setting this bit initiates the cache command indicated by bits 27-24. Reading this bit indicates if a command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. 0 Write: no effect. Read: no cache command active. 1 Write: initiate command indicated by bits 27-24. Read: cache command active.
30–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 PUSHW1	Push Way 1

Table continues on the next page...

LMEM_PCCCR field descriptions (continued)

Field	Description
	0 No operation 1 When setting the GO bit, push all modified lines in way 1
26 INVW1	Invalidate Way 1 **NOTE:** If the PUSHW1 and INVW1 bits are set, then after setting the GO bit, push all modified lines in way 1 and invalidate all lines in way 1 (clear way 1). 0 No operation 1 When setting the GO bit, invalidate all lines in way 1
25 PUSHW0	Push Way 0 0 No operation 1 When setting the GO bit, push all modified lines in way 0
24 INVW0	Invalidate Way 0 **NOTE:** If the PUSHW0 and INVW0 bits are set, then after setting the GO bit, push all modified lines in way 0 and invalidate all lines in way 0 (clear way 0). 0 No operation 1 When setting the GO bit, invalidate all lines in way 0.
23–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 PCCR3	Forces no allocation on cache misses (must also have ACCR2 asserted)
2 PCCR2	Forces all cacheable spaces to write through
1 ENWRBUF	Enable Write Buffer 0 Write buffer disabled 1 Write buffer enabled
0 ENCACHE	Cache enable 0 Cache disabled 1 Cache enabled

29.2.2 Cache line control register (LMEM_PCCLCR)

This register defines specific line-sized cache operations to be performed using a specific cache line address or a physical address.

If a physical address is specified, both ways of the cache are searched, and the command is only performed on the way which hits.

Address: E008_2000h base + 4h offset = E008_2004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				LACC	LADSEL	LCMD		0	LCWAY	LCIMB	LCIVB	0			TDSEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	WSEL	0	CACHEADDR											0	LGO
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PCCLCR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 LACC	Line access type 0 Read 1 Write
26 LADSEL	Line Address Select When using the cache address, the way must also be specified in CLCR[WSEL]. When using the physical address, both ways are searched and the command is performed only if a hit. 0 Cache address 1 Physical address
25–24 LCMD	Line Command 00 Search and read or write 01 Invalidate 10 Push 11 Clear

Table continues on the next page...

LMEM_PCCLCR field descriptions (continued)

Field	Description
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 LCWAY	Line Command Way Indicates the way used by the line command.
21 LCIMB	Line Command Initial Modified Bit If command used cache address and way, then this bit shows the initial state of the modified bit If command used physical address and a hit, then this bit shows the initial state of the modified bit. If a miss, this bit reads zero.
20 LCIVB	Line Command Initial Valid Bit If command used cache address and way, then this bit shows the initial state of the valid bit If command used physical address and a hit, then this bit shows the initial state of the valid bit. If a miss, this bit reads zero.
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 TDSEL	Tag/Data Select Selects tag or data for search and read or write commands. 0 Data 1 Tag
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 WSEL	Way select Selects the way for line commands. 0 Way 0 1 Way 1
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–2 CACHEADDR	Cache address CLCR[11:4] bits are used to access the tag arrays CLCR[11:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CSAR[LGO] 0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits 27-24. Read: line command active.

29.2.3 Cache search address register (LMEM_PCCSAR)

The CSAR register is used to define the explicit cache address or the physical address for line-sized commands specified in the CLCR[LADSEL] bit.

Address: E008_2000h base + 8h offset = E008_2008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHYADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHYADDR														0	LGO
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PCCSAR field descriptions

Field	Description
31–2 PHYADDR	Physical Address PHYADDR represents bits [31:2] of the system address. CSAR[31:12] bits are used for tag compare CSAR[11:4] bits are used to access the tag arrays CSAR[11:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CLCR[LGO] 0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits CLCR[27:24]. Read: line command active.

29.2.4 Cache read/write value register (LMEM_PCCCVR)

The CCVR register is used to source write data or return read data for the commands specified in the CLCR register.

Address: E008_2000h base + Ch offset = E008_200Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PCCCVR field descriptions

Field	Description
31–0 DATA	Cache read/write Data For tag search, read or write: • CCVR[31:12] bits are used for tag array R/W value • CCVR[11:4] bits are used for tag set address on reads; unused on writes • CCVR[3:2] bits are reserved For data search, read or write: • CCVR[31:0] bits are used for data array R/W value

29.2.5 Cache control register (LMEM_PSCCR)

Address: E008_2000h base + 800h offset = E008_2800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GO	0			PUSHW1	INVW1	PUSHW0	INVW0	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														ENWRBUF	ENCACHE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PSCCR field descriptions

Field	Description
31 GO	Initiate Cache Command Setting this bit initiates the cache command indicated by bits 27-24. Reading this bit indicates if a command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. 0 Write: no effect. Read: no cache command active. 1 Write: initiate command indicated by bits 27-24. Read: cache command active.
30–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 PUSHW1	Push Way 1 0 No operation 1 When setting the GO bit, push all modified lines in way 1
26 INVW1	Invalidate Way 1 **NOTE:** If the PUSHW1 and INVW1 bits are set, then after setting the GO bit, push all modified lines in way 1 and invalidate all lines in way 1 (clear way 1). 0 No operation 1 When setting the GO bit, invalidate all lines in way 1
25 PUSHW0	Push Way 0 0 No operation 1 When setting the GO bit, push all modified lines in way 0
24 INVW0	Invalidate Way 0 **NOTE:** If the PUSHW0 and INVW0 bits are set, then after setting the GO bit, push all modified lines in way 0 and invalidate all lines in way 0 (clear way 0). 0 No operation 1 When setting the GO bit, invalidate all lines in way 0.
23–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 ENWRBUF	Enable Write Buffer 0 Write buffer disabled 1 Write buffer enabled
0 ENCACHE	Cache enable 0 Cache disabled 1 Cache enabled

29.2.6 Cache line control register (LMEM_PSCLCR)

This register defines specific line-sized cache operations to be performed using a specific cache line address or a physical address.

If a physical address is specified, both ways of the cache are searched, and the command is only performed on the way which hits.

Address: E008_2000h base + 804h offset = E008_2804h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				LACC	LADSEL	LCMD		0	LCWAY	LCIMB	LCIVB	0			TDSEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	WSEL	0	CACHEADDR											0	LGO
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PSCLCR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 LACC	Line access type 0 Read 1 Write
26 LADSEL	Line Address Select When using the cache address, the way must also be specified in CLCR[WSEL].

Table continues on the next page...

LMEM_PSCLCR field descriptions (continued)

Field	Description
	When using the physical address, both ways are searched and the command is performed only if a hit. 0 Cache address 1 Physical address
25–24 LCMD	Line Command 00 Search and read or write 01 Invalidate 10 Push 11 Clear
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 LCWAY	Line Command Way Indicates the way used by the line command.
21 LCIMB	Line Command Initial Modified Bit If command used cache address and way, then this bit shows the initial state of the modified bit If command used physical address and a hit, then this bit shows the initial state of the modified bit. If a miss, this bit reads zero.
20 LCIVB	Line Command Initial Valid Bit If command used cache address and way, then this bit shows the initial state of the valid bit If command used physical address and a hit, then this bit shows the initial state of the valid bit. If a miss, this bit reads zero.
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 TDSEL	Tag/Data Select Selects tag or data for search and read or write commands. 0 Data 1 Tag
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 WSEL	Way select Selects the way for line commands. 0 Way 0 1 Way 1
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–2 CACHEADDR	Cache address CLCR[11:4] bits are used to access the tag arrays CLCR[11:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

LMEM_PSCLCR field descriptions (continued)

Field	Description
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CSAR[LGO] 0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits 27-24. Read: line command active.

29.2.7 Cache search address register (LMEM_PSCSAR)

The CSAR register is used to define the explicit cache address or the physical address for line-sized commands specified in the CLCR[LADSEL] bit.

Address: E008_2000h base + 808h offset = E008_2808h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHYADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHYADDR														0	LGO
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PSCSAR field descriptions

Field	Description
31–2 PHYADDR	Physical Address PHYADDR represents bits [31:2] of the system address. CSAR[31:12] bits are used for tag compare CSAR[11:4] bits are used to access the tag arrays CSAR[11:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CLCR[LGO]

Table continues on the next page...

LMEM_PSCSAR field descriptions (continued)

Field	Description
	0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits CLCR[27:24]. Read: line command active.

29.2.8 Cache read/write value register (LMEM_PSCCVR)

The CCVR register is used to source write data or return read data for the commands specified in the CLCR register.

Address: E008_2000h base + 80Ch offset = E008_280Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PSCCVR field descriptions

Field	Description
31–0 DATA	Cache read/write Data For tag search, read or write: • CCVR[31:12] bits are used for tag array R/W value • CCVR[11:4] bits are used for tag set address on reads; unused on writes • CCVR[3:2] bits are reserved For data search, read or write: • CCVR[31:0] bits are used for data array R/W value

29.3 Functional Description

29.3.1 LMEM Function

The Local Memory Controller receives the following requests:

- Core master bus requests on the Processor Code (PC) bus,
- Core master bus requests on the Processor Space (PS) bus, and
- SRAM controller requests from all other bus masters on the backdoor port.

The Local Memory Controller address decode logic routes these accesses and also provides any crossbar switch slave target logic. Finally, the Local Memory controller provides the needed MPU connections for checking all SRAM controller and cacheable accesses.

The programming model for the Code and System Caches is accessed via the core's Private Peripheral Bus (PPB).

29.3.1.1 Processor Code accesses

Processor Code accesses are routed to the SRAM_L if they are mapped to that space. All other PC accesses are routed to the Code Cache Memory Controller. This controller then processes the cacheable accesses as needed, while bypassing the non-cacheable, cache write-through, cache miss, and cache maintenance accesses to the CCM bus and the crossbar switch using the Master0 port.

29.3.1.2 Processor Space accesses

Processor Space accesses are routed to the SRAM_U if they are mapped to that space. All other PS accesses are routed to the PS Cache Memory Controller. This controller then processes the cacheable accesses as needed, while bypassing the non-cacheable, cache write-through, cache miss, and cache maintenance accesses to the CCM bus and the crossbar switch using the Master1 port.

29.3.1.3 Backdoor port accesses

All LMEM backdoor port accesses are for the SRAM controller. These accesses go to the SRAM_L or the SRAM_U depending on their specific address.

29.3.2 SRAM Function

29.3.2.1 SRAM Configuration

The figure below shows how the SRAM controller is configured.

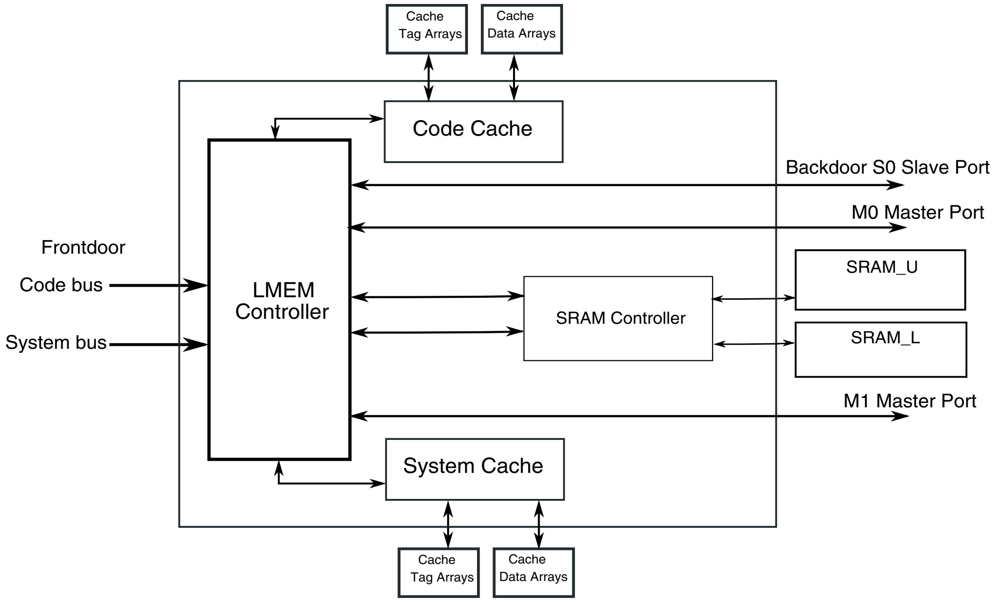


Figure 29-10. SRAM Configuration

29.3.2.2 SRAM Arrays

The on-chip SRAM is split into two logical arrays, SRAM_L and SRAM_U.

From equal-sized memories, valid address ranges for SRAM_L and SRAM_U are then defined as:

- SRAM_L = 0x1F80_0000 - (0x1F80_0000 + SRAM_size/2)
- SRAM_U = 0x3F80_0000 - (0x3F80_0000 + SRAM_size/2)

29.3.2.3 SRAM accesses

The SRAM is split into two logical arrays that are 64-bits wide:

- SRAM_L — Accessible by the code bus of the Cortex-M4 core and by the backdoor port.
- SRAM_U — Accessible by the system bus of the Cortex-M4 core and by the backdoor port.

The backdoor port makes the SRAM accessible to the non-core bus masters (such as DMA).

Figure 29-11 illustrates the SRAM accesses within the device.

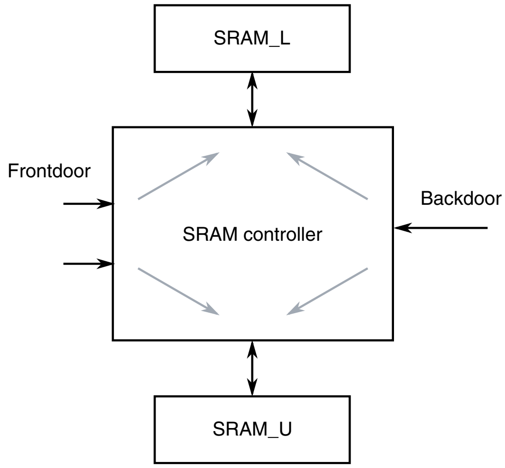


Figure 29-11. SRAM access diagram

The following simultaneous accesses can be made to different logical halves of the SRAM:

- Core code and core system
- Core code and non-core master
- Core system and non-core master

NOTE

Two non-core masters cannot access SRAM simultaneously. The required arbitration and serialization is provided by the crossbar switch. The SRAM_{L,U} arbitration is controlled by the SRAM controller based on the configuration bits in the MCM module.

29.3.3 Cache Function

The caches on this device are structured as follows. Both caches have a 2-way set-associative cache structure with a total size of 32 KBytes. The caches have a 32-bit address, 64-bit data paths and a 32-byte line size. The cache tags and data storage use single-port, synchronous RAMs.

For these 16-KByte caches, each cache TAG function uses two 256 x 22-bit RAM arrays and the cache DATA function uses two 1024 x 32-bit RAM arrays. The cache TAG entries store 20 bits of upper address as well as a modified and valid bit per cache line. The cache DATA entries store eight byes of code or data.

All normal cache accesses use physical addresses. This leads to the following cache address use:

CACHE - 16 KByte size = (256 sets) x (32-byte lines) x (2-way set associative)

TAG:

- address[31:13] used in tag for compare (hit) logic
- address[12:5] used to select 1 of 256 sets
- address[3:0] not used

DATA

- address[31:13] not used
- address[12:5] used to select one of 256 sets
- address[4:2] used to select one of eight 32-bit words within a set
- address[1:0] used to select the byte within the 32-bit word

29.3.4 Cache Control

The Code and System Caches are disabled at reset. Cache tag and data arrays are not cleared at reset. Therefore, to enable the caches, cache commands must be done to clear and initialize the required tag array bits and to configure and enable the caches.

29.3.4.1 Cache set commands

The cache set commands may operate on:

- all of way 0,
- all of way 1, or
- all of both ways (complete cache).

Cache set commands are initiated using the upper bits in the CCR register. Cache set commands perform their operation on the cache independent of the cache enable bit, CCR[ENCACHE].

A cache set command is initiated by setting the CCR[GO] bit. This bit also acts as a busy bit for set commands. It stays set while the command is active and is cleared by the hardware when the set command completes.

Supported cache set commands are given in Table 29-11. Set commands work as follows:

- Invalidate – Unconditionally clear valid and modify bits of a cache entry.
- Push – Push a cache entry if it is valid and modified, then clear the modify bit. If entry not valid or not modified, leave as is.
- Clear – Push a cache entry if it is valid and modified, then clear the valid and modify bits. If entry not valid or not modified, clear the valid bit.

Table 29-11. Cache Set Commands

CCR[27:24]				Command
PUSH W1	INVW1	PUSH W0	INVW0	
0	0	0	0	NOP
0	0	0	1	Invalidate all way 0
0	0	1	0	Push all way 0
0	0	1	1	Clear all way 0
0	1	0	0	Invalidate all way 1
0	1	0	1	Invalidate all way 1; invalidate all way 0 (invalidate cache)
0	1	1	0	Invalidate all way 1; push all way 0
0	1	1	1	Invalidate all way 1; clear all way 0
1	0	0	0	Push all way 1
1	0	0	1	Push all way 1; invalidate all way 0
1	0	1	0	Push all way 1; push all way 0 (push cache)
1	0	1	1	Push all way 1; clear all way 0
1	1	0	0	Clear all way 1
1	1	0	1	Clear all way 1; invalidate all way 0
1	1	1	0	Clear all way 1; push all way 0
1	1	1	1	Clear all way 1; clear all way 0 (clear cache)

After a reset, complete an invalidate cache command before using the cache. It is possible to combine the cache invalidate command with the cache enable. That is, setting CCR to 0x8500_0003 will invalidate the cache and enable the cache and write buffer.

29.3.4.2 Cache line commands

Cache line commands operate on a single line in the cache at a time. Cache line commands can be performed using a physical or cache address.

- A cache address consists of a set address and a way select. The line command acts on the specified cache line.
- Cache line commands with physical addresses first search both ways of the cache set specified by bits [11:4] of the physical address. If they hit, the commands perform their action on the hit way.

Cache line commands are specified using the upper bits in the CLCR register. Cache line commands perform their operation on the cache independent of the cache enable bit (CCR[ENCACHE]). Using a cache address, the command can be completely specified using the CLCR register. Using a physical address, the command must also use the CSAR register to specify the physical address.

A line cache command is initiated by setting the line command go bit (CLCR[LGO] or CSAR[LGO]). This bit also acts a a busy bit for line commands. It stays set while the command is active and is cleared by the hardware when the command completes.

The CLCR[27:24] bits select the line command as follows:

Table 29-12. Cache Line Commands

CLCR[27:24]			Command
LACC	LADSEL	LCMD	
0	0	00	Search by cache address and way
0	0	01	Invalidate by cache address and way
0	0	10	Push by cache address and way
0	0	11	Clear by cache address and way
0	1	00	Search by physical address
0	1	01	Invalidate by physical address
0	1	10	Push by physical address
0	1	11	Clear by physical address
1	0	00	Write by cache address and way
1	0	01	Reserved, NOP
1	0	10	Reserved, NOP
1	0	11	Reserved, NOP
1	1	*xx*	Reserved, NOP

29.3.4.2.1 Executing a series of line commands using cache addresses

A series of line commands with incremental cache addresses can be performed by just writing to the CLCR.

- Place the command in CLCR[27:24],
- Set the way (CLCR[WSEL]) and tag/data (CLCR[TDSEL]) controls as needed,
- Place the cache address in CLCR[CACHEADDR], and
- Set the line command go bit (CLCR[LGO]).

When one line command completes, initiate the next command by following these steps:

- Increment the cache address (at bit 2 to step through data or at bit 4 to step through lines), and
- Set the line command go bit (CLCR[LGO]).

29.3.4.2.2 Executing a series of line commands using physical addresses

Perform a series of line commands with incremental physical addresses using the following steps:

- Write to the CLCR.
  - Place the command in CLCR[27:24]
  - Set the tag/data (CLCR[TDSEL]) control
- Place the physical address in CSAR[PHYADDR] and set the line command go bit (CSAR[LGO]).

When one line command completes, initiate the next command by following these steps:

- Increment the physical address (at bit 2 to step through data or at bit 4 to step through lines), and
- Set the line command go bit (CSAR[LGO]).

The line command go bit is shared between the CLCR and CSAR registers, so that the above steps can be completed in a single write to the CSAR register.

29.3.4.2.3 Line command results

At completion of a line command, the CLCR register contains information on the initial state of the line targeted by the command. For line commands with cache addresses, this information is read before the line command action is performed from the targeted cache line. For line commands with physical addresses, this information is read on a hit before the line command action is performed from the hit cache line or has initial valid bit cleared if the command misses. In general, if the valid indicator (CLCR[LCIVB] is cleared, the targeted line was invalid at the start of the line command and no line operation was performed.

Table 29-13. Line command results

CLCR[22:20]			For cache address commands	For physical address commands
LCWAY	LCIMB	LCIVB		
0	0	0	Way 0 line was invalid	No hit
0	0	1	Way 0 valid, not modified	Way 0 valid, not modified
0	1	0	Way 0 line was invalid	No hit
0	1	1	Way 0 valid and modified	Way 0 valid and modified
1	0	0	Way 1 line was invalid	No hit
1	0	1	Way 1 valid, not modified	Way 1 valid, not modified
1	1	0	Way 1 line was invalid	No hit
1	1	1	Way 1 valid and modified	Way 1 valid and modified

At completion of a line command other than a write, the CCVR (Cache R/W Value Register) contains information on the initial state of the line tag or data targeted by the command. For line commands, CLCR[TDSEL] selects between tag and data. If the line command used a physical address and missed, the data is don't care. For write commands, the CCVR holds the write data.

Chapter 30
Quad Serial Peripheral Interface (QuadSPI)

30.1 Introduction

The Quad Serial Peripheral Interface (QuadSPI) block acts as an interface to one single or two external serial flash devices, each with up to four bidirectional data lines.

30.1.1 Features

The QuadSPI supports the following features:

- Flexible sequence engine to support various flash vendor devices.
- Single, dual, quad and octal mode of operation.
- DDR/DTR mode wherein the data is generated on every edge of the serial flash clock.
- Support for flash data strobe signal for data sampling in DDR and SDR mode.
- Support for parallel writes via register mapped interface in single io mode.
- Two identical serial flash devices can be connected and accessed in parallel for data read operations, forming one (virtual) flash memory with doubled readout bandwidth.
- DMA support to read RX Buffer data via AMBA AHB bus (64-bit width interface) or IP registers space (32-bit access).
    - Inner loop size of DMA access can be configured.
- Multi master accesses with priority
    - Flexible and configurable buffer for each master
- Fourteen interrupt conditions (see Table 30-425)

- Memory mapped read access to connected flash devices.
- Programmable sequence engine to cater to future command/protocol changes and able to support all existing vendor commands and operations.
    - Supports 3-byte and 4-byte addressing.

30.1.2 Block Diagram

The following figure is a block diagram of the Quad Serial Peripheral Interface (QuadSPI) module.

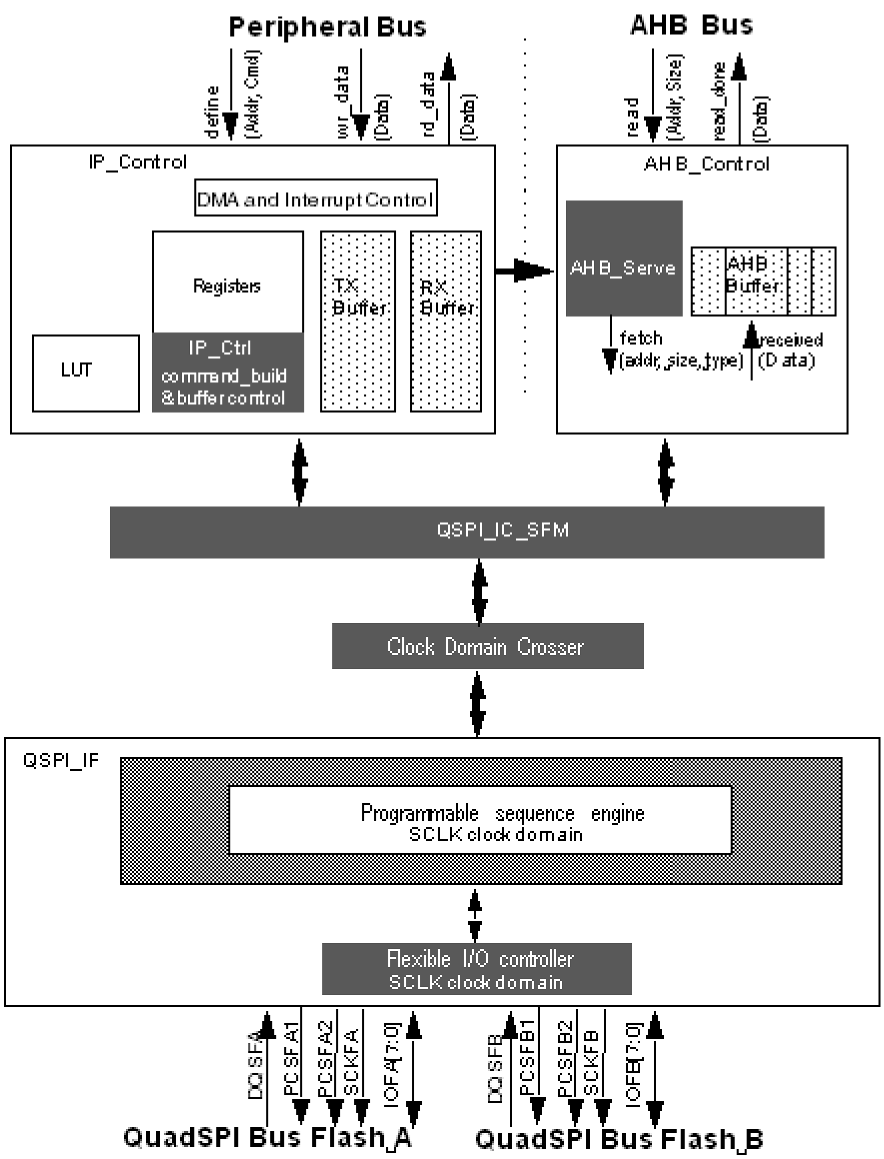


Figure 30-1. QuadSPI Block Diagram

30.1.3 QuadSPI Modes of Operation

This section provides information about the modes in which the QuadSPI module can be used.

30.1.3.1 Normal Mode

In this mode one or two external serial flash memory device can be accessed. Further details about this mode of operation can be found in Modes of Operation(Normal Mode) section.

30.1.3.2 Module Disable Mode

The Module Disable Mode is used for power management of the device containing the QuadSPI module, it is controlled by signals external to the QuadSPI. The clock to the non-memory mapped logic in the QuadSPI can be stopped while in the Module Disable Mode.

30.1.3.3 Stop Mode

The Stop Mode is also used for power management. When a request is made to enter Stop Mode, the QuadSPI block completes the action currently being processed, before the request is acknowledged.

30.1.4 Acronyms and Abbreviations

The following table contains acronyms and abbreviations used in this document.

Table 30-1. Acronyms and Abbreviations

Terms	Description
AHB	Advanced High-performance Bus, version of AMBA
AMBA	Advanced Microcontroller Bus Architecture
BE	Big Endian Byte Ordering
CS	Chip Select.
DMA	Direct Memory Access.
MB	Megabyte. Each MB is 1024 * 1024 bytes

Table continues on the next page...

Table 30-1. Acronyms and Abbreviations (continued)

Terms	Description
IFM	Individual Flash Mode
PFM	Parallel Flash Mode
LSB	Least Significant Bit
MSB	Most Significant Bit
PCS	Peripheral Chip Select
QSPI, QuadSPI	Quad Serial Peripheral Interface
SCK	Serial Communications Clock
w1c	Write 1 to clear, writing a '1' to this field resets the flag

30.1.5 Glossary for QuadSPI module

Table 30-2. Glossary

Term	Definition
AHB Command	An AHB Command is a SFM Command triggered by a read access to the address range belonging to the memory mapped access defined in Table 30-388. Refer to AHB Commands for details.
Asserted	A signal that is asserted is in its active state. An active low signal changes from logic level one to logic level zero when asserted, and an active high signal changes from logic level zero to logic level one.
Clear	To clear a bit or bits means to establish logic level zero on the bit or bits.
Clock Phase	Determines when the data should be sampled relative to the active edge of SCK
Clock Polarity	Determines the idle state of the SCK signal.
Drain	To remove entries from a FIFO by software or hardware.
Endianess	Byte Ordering scheme.
Field	Two or more register bits grouped together.
Fill	To add entries to a FIFO by software or hardware.
Host	Refers to another functional block in the device containing the QuadSPI module
Instruction Code	8 bits defining the type of command to be executed.
IP Command	An IP Command is a SFM Command triggered by writing into the QSPI_IPCR[SEQID] field.
Logic level one	The voltage that corresponds to Boolean true (1) state.
Logic level zero	The voltage that corresponds to Boolean false (0) state.
Negated	A signal that is negated is in its inactive state. An active low signal changes from logic level '0' to logic level '1' when negated, and an active high signal changes from logic level '1' to logic level '0'.
QSPI_AMBA_BASE	First address of QuadSPI address space on system memory map.
QSPI_ARDB_BASE	First address of QuadSPI Rx Buffer on system memory map.
Set	To set a bit or bits means to establish logic level one on the bit or bits.

Table continues on the next page...

Table 30-2. Glossary (continued)

Term	Definition
RX Buffer PUSH Event	Addition of valid entries into the RX Buffer. In the default case each Buffer PUSH Event adds 2 entries to the RX Buffer since the interface to the serial clock domain is 64 bits in width. Depending on the number of bytes read from the serial flash device it is possible for the very last Buffer PUSH Event that only one entry is added. The QSPI_RBSR[RDBFL] field is incremented by the number of entries added to the RX Buffer.
RX Buffer POP Event	Removal of valid entries from the RX Buffer. Each Buffer POP Event removes (QSPI_RBCT[WMRK] + 1) valid entries from the buffer. The QSPI_RBSR[RDBFL] field is decremented by the same number and the QSPI_RBSR[RDCTR] field is incremented accordingly.
Individual Flash Mode	Access to a single, individual serial flash device. Refer to Serial Flash Access Schemes for details.
Parallel Flash Mode	Read access to two serial flash devices attached to the QuadSPI module in parallel. Refer to Serial Flash Access Schemes for details.
SFM Command	Serial Flash Memory Command. A SFM command consists of an instruction code and all other parameters (e.g. size or mode bytes) needed for that specific instruction code. Triggering a command either initiates a transaction on the external serial flash or results in an error. Refer to Table 30-436 for details on errors.
Single/Dual/Quad Instructions	Depending from the serial flash device connected to the QuadSPI module there will be instructions using a different number of data lines. • Single: Single line I/O with one data out and one data in line to/from the serial flash device. • Dual: Dual line I/O with two bidirectional I/O lines, driven alternatively by the serial flash device or the QuadSPI module • Quad: Quad line I/O with 4 bidirectional I/O lines, driven alternatively by the serial flash device or the QuadSPI
Transaction	A transaction consists of all flags, data and signals in either direction to execute a command for an attached serial flash device. It is a combination of chip select, sclk, instruction code, address, mode- and/or dummy bytes, transmit and/or receive data.
LUT	Look-up-table.

30.2 External Signal Description

This section provides the external signal information of the QuadSPI module.

The following table lists the external signals belonging to the QuadSPI, module in conjunction with the different modes of operation.

Table 30-3. Signal Properties

Signal Name	Function	Direction	Description
PCSFA1	Peripheral Chip Select Flash A1	O	This signal is the chip select for the serial flash device A1. A1 represents the first device in a dual-die package flash A or the first of the two flash devices that share IOFA. Refer to Dual Die Flashes for more details.
PCSFA2	Peripheral Chip Select Flash A2	O	This signal is the chip select for the serial flash device A2. A2 represents the second device in a dual-die package flash A or the second of the two flash devices that share IOFA. Refer to Dual Die Flashes for more details.

Table continues on the next page...

Table 30-3. Signal Properties (continued)

Signal Name	Function	Direction	Description
PCSFB1	Peripheral Chip Select Flash B1	O	This signal is the chip select for the serial flash device B1. B1 represents the first device in a dual-die package flash B or the first of the two flash devices that share IOFB. Refer to Dual Die Flashes for more details.
PCSFB2	Peripheral Chip Select Flash B2	O	This signal is the chip select for the serial flash device B2. B2 represents the second device in a dual-die package flash B or the second of the two flash devices that share IOFB. Refer to Dual Die Flashes for more details.
SCKFA	Serial Clock Flash A	O	This signal is the serial clock output to the serial flash device A.
SCKFB	Serial Clock Flash B	O	This signal is the serial clock output to the serial flash device B.
IOFA[7:0]	Serial I/O Flash A	I/O	These signals are the data I/O lines to/from the serial flash device A. Refer to Driving External Signals for details about the signal drive and timing behavior. Note that the signal pins of the serial flash device may change their function according to the SFM Command executed, leaving them as control inputs when Single and Dual Instructions are executed. The QuadSPImodule supports driving these inputs to dedicated values. During the execution of Single and Dual Instructions, the logic level which is applied to the IOFA[3] and IOFA[2] output signals, is determined by the QSPI_MCR[ISD3FA] and the QSPI_MCR[ISD2FA] bits.
IOFB[7:0]	Serial I/O Flash B	I/O	These signals are the data I/O lines to/from the serial flash device B. Refer to Driving External Signals for details about the signal drive and timing behavior. Note that the signal pins of the serial flash device may change their function according to the SFM Command executed, leaving them as control inputs when Single and Dual Instructions are executed. The QuadSPImodule supports driving these inputs to dedicated values. During the execution of Single and Dual Instructions the logic level which is applied to the IOFB[3] and IOFB[2] output signals is determined by the QSPI_MCR[ISD3FB] and the QSPI_MCR[ISD2FB] bits.
DQSFA	Data Strobe signal Flash A	I	Data strobe signal for port A. Some flash vendors provide the DQS signal to which the read data is aligned in DDR mode.
DQSFB	Data Strobe signal Flash B	I	Data strobe signal for port B. Some flash vendors provide the DQS signal to which the read data is aligned in DDR mode.

30.2.1 Driving External Signals

The different phases of serial flash access scheme are shown in the following figure.

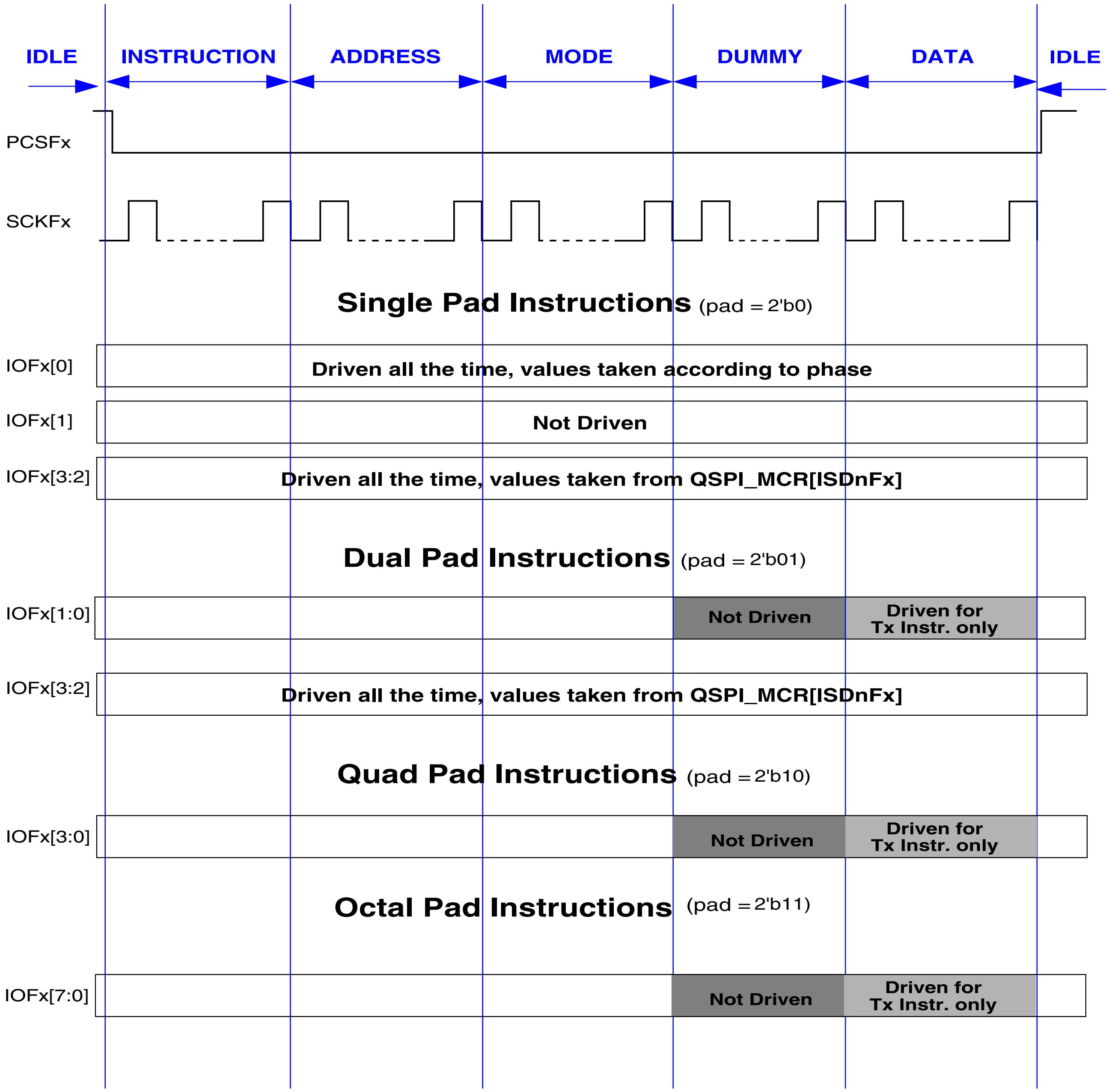


Figure 30-2. Serial Flash Access Scheme

The different phases and the I/O driving characteristics of the QuadSPI module are characterized in the following way:

- IDLE: Serial flash device not selected. No interaction with the serial flash device. All IOFx signals driven.
- INSTRUCTION: Serial flash device selected. The instruction is sent to the serial flash device. All IOFx signals are driven.

- ADDRESS: Serial Flash Address is sent to the device. All IOFx signals are driven. Note that this phase is not applicable for all SFM Commands.
- MODE: Mode bytes are sent to the serial flash device. All IOFx signals are driven. Note that this phase is not applicable for all SFM Commands.
- DUMMY: Dummy clocks are provided to the serial flash device. Refer to the Figure 30-2 for the IOFx signals driven. The actual data lines required for the SFM Command executed are not driven for data read commands. Note that this phase is not applicable for all SFM Commands.
- DATA: Serial flash data are sent to or received from the serial flash device. Refer to the preceding figure for the IOFx signals driven. The actual data lines required for the SFM Command executed are not driven for data read commands. Note that this phase is not applicable for all SFM Commands.

The PCSFx and SCKFx signals are driven permanently throughout all the phases.

In Individual Flash Mode this applies to the selected flash device. In Parallel Flash Mode this applies to both serial flash devices simultaneously.

30.3 Memory Map and Register Definition

This section provides the memory map and register definitions of the QuadSPI module.

30.3.1 Register Write Access

This section describes the write access restriction terms that apply to all registers, which can be one of the following:

- **Register Write Access Restriction**

  For each register bit and register field, the write access conditions are specified in the detailed register description. A description of the write access conditions is given in the following table. If, for a specific register bit or field, none of the given write access conditions is fulfilled, any write attempt to this register bit or field is ignored without any notification. The values of the bits or fields are not changed.

  The condition term [A or B] indicates that the register or field can be written to if at least one of the conditions is fulfilled.

Table 30-4. Register Write Access Restrictions

Condition	Description
Anytime	No write access restriction.
Disabled Mode	Write access only if *QSPI_MCR[MDIS] = 1*.
Normal Mode	Write access only if the module is in *Normal Mode*.

- **Register Write Access Requirements**

  All registers can be accessed with 8-bit, 16-bit, and 32-bit wide operations. For some of the registers, at least a 16/32-bit wide write access is required to ensure correct operation. This write access requirement is stated in the detailed register description for each register affected.

30.3.2 Peripheral Bus Register Descriptions

This section provides the peripheral bus register information of the QuadSPI module. For register diagram and description table, please refer to Section "Peripheral Bus Memory Map and Registers".

This section provides the memory map and register definitions of the QuadSPI module.

QuadSPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_4000	Module Configuration Register (QuadSPI0_MCR)	32	R/W	000F_4000h	30.32.1/ 1207
4004_4008	IP Configuration Register (QuadSPI0_IPCR)	32	R/W	0000_0000h	30.32.2/ 1208
4004_400C	Flash Configuration Register (QuadSPI0_FLSHCR)	32	R/W	0000_0303h	30.32.3/ 1209
4004_4010	Buffer0 Configuration Register (QuadSPI0_BUF0CR)	32	R/W	0000_0000h	30.32.4/ 1210
4004_4014	Buffer1 Configuration Register (QuadSPI0_BUF1CR)	32	R/W	0000_0000h	30.32.5/ 1211
4004_4018	Buffer2 Configuration Register (QuadSPI0_BUF2CR)	32	R/W	0000_0000h	30.32.6/ 1211
4004_401C	Buffer3 Configuration Register (QuadSPI0_BUF3CR)	32	R/W	See section	30.32.7/ 1212

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_4020	Buffer Generic Configuration Register (QuadSPI0_BFGENCR)	32	R/W	0000_0000h	30.32.8/ 1213
4004_4030	Buffer0 Top Index Register (QuadSPI0_BUF0IND)	32	R/W	0000_0000h	30.32.9/ 1214
4004_4034	Buffer1 Top Index Register (QuadSPI0_BUF1IND)	32	R/W	0000_0000h	30.32.10/ 1215
4004_4038	Buffer2 Top Index Register (QuadSPI0_BUF2IND)	32	R/W	0000_0000h	30.32.11/ 1215
4004_4100	Serial Flash Address Register (QuadSPI0_SFAR)	32	R/W	0000_0000h	30.32.12/ 1216
4004_4108	Sampling Register (QuadSPI0_SMPR)	32	R/W	0000_0000h	30.32.13/ 1217
4004_410C	RX Buffer Status Register (QuadSPI0_RBSR)	32	R	0000_0000h	30.32.14/ 1218
4004_4110	RX Buffer Control Register (QuadSPI0_RBCT)	32	R/W	0000_0000h	30.32.15/ 1219
4004_4150	TX Buffer Status Register (QuadSPI0_TBSR)	32	R	0000_0000h	30.32.16/ 1220
4004_4154	TX Buffer Data Register (QuadSPI0_TBDR)	32	R/W	0000_0000h	30.32.17/ 1221
4004_415C	Status Register (QuadSPI0_SR)	32	R	0000_3800h	30.32.18/ 1222
4004_4160	Flag Register (QuadSPI0_FR)	32	w1c	0800_0000h	30.32.19/ 1225
4004_4164	Interrupt and DMA Request Select and Enable Register (QuadSPI0_RSER)	32	R/W	0000_0000h	30.32.20/ 1228
4004_4168	Sequence Suspend Status Register (QuadSPI0_SPNDST)	32	R	0000_0000h	30.32.21/ 1231
4004_416C	Sequence Pointer Clear Register (QuadSPI0_SPTRCLR)	32	R/W	0000_0000h	30.32.22/ 1233
4004_4180	Serial Flash A1 Top Address (QuadSPI0_SFA1AD)	32	R/W	0000_0000h	30.32.23/ 1233
4004_4184	Serial Flash A2 Top Address (QuadSPI0_SFA2AD)	32	R/W	0000_0000h	30.32.24/ 1234
4004_4188	Serial Flash B1Top Address (QuadSPI0_SFB1AD)	32	R/W	0000_0000h	30.32.25/ 1234
4004_418C	Serial Flash B2Top Address (QuadSPI0_SFB2AD)	32	R/W	0000_0000h	30.32.26/ 1235
4004_4200	RX Buffer Data Register (QuadSPI0_RBDR0)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4204	RX Buffer Data Register (QuadSPI0_RBDR1)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4208	RX Buffer Data Register (QuadSPI0_RBDR2)	32	R/W	0000_0000h	30.32.27/ 1235

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_420C	RX Buffer Data Register (QuadSPI0_RBDR3)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4210	RX Buffer Data Register (QuadSPI0_RBDR4)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4214	RX Buffer Data Register (QuadSPI0_RBDR5)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4218	RX Buffer Data Register (QuadSPI0_RBDR6)	32	R/W	0000_0000h	30.32.27/ 1235
4004_421C	RX Buffer Data Register (QuadSPI0_RBDR7)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4220	RX Buffer Data Register (QuadSPI0_RBDR8)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4224	RX Buffer Data Register (QuadSPI0_RBDR9)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4228	RX Buffer Data Register (QuadSPI0_RBDR10)	32	R/W	0000_0000h	30.32.27/ 1235
4004_422C	RX Buffer Data Register (QuadSPI0_RBDR11)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4230	RX Buffer Data Register (QuadSPI0_RBDR12)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4234	RX Buffer Data Register (QuadSPI0_RBDR13)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4238	RX Buffer Data Register (QuadSPI0_RBDR14)	32	R/W	0000_0000h	30.32.27/ 1235
4004_423C	RX Buffer Data Register (QuadSPI0_RBDR15)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4240	RX Buffer Data Register (QuadSPI0_RBDR16)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4244	RX Buffer Data Register (QuadSPI0_RBDR17)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4248	RX Buffer Data Register (QuadSPI0_RBDR18)	32	R/W	0000_0000h	30.32.27/ 1235
4004_424C	RX Buffer Data Register (QuadSPI0_RBDR19)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4250	RX Buffer Data Register (QuadSPI0_RBDR20)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4254	RX Buffer Data Register (QuadSPI0_RBDR21)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4258	RX Buffer Data Register (QuadSPI0_RBDR22)	32	R/W	0000_0000h	30.32.27/ 1235
4004_425C	RX Buffer Data Register (QuadSPI0_RBDR23)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4260	RX Buffer Data Register (QuadSPI0_RBDR24)	32	R/W	0000_0000h	30.32.27/ 1235

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_4264	RX Buffer Data Register (QuadSPI0_RBDR25)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4268	RX Buffer Data Register (QuadSPI0_RBDR26)	32	R/W	0000_0000h	30.32.27/ 1235
4004_426C	RX Buffer Data Register (QuadSPI0_RBDR27)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4270	RX Buffer Data Register (QuadSPI0_RBDR28)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4274	RX Buffer Data Register (QuadSPI0_RBDR29)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4278	RX Buffer Data Register (QuadSPI0_RBDR30)	32	R/W	0000_0000h	30.32.27/ 1235
4004_427C	RX Buffer Data Register (QuadSPI0_RBDR31)	32	R/W	0000_0000h	30.32.27/ 1235
4004_4300	LUT Key Register (QuadSPI0_LUTKEY)	32	R/W	5AF0_5AF0h	30.32.28/ 1236
4004_4304	LUT Lock Configuration Register (QuadSPI0_LCKCR)	32	R/W	0000_0002h	30.32.29/ 1237
4004_4310	Look-up Table register (QuadSPI0_LUT0)	32	R/W	0818_0403h	30.32.30/ 1238
4004_4314	Look-up Table register (QuadSPI0_LUT1)	32	R/W	2400_1C08h	30.32.31/ 1239
4004_4318	Look-up Table register (QuadSPI0_LUT2)	32	R/W	0000_0000h	30.32.32/ 1240
4004_431C	Look-up Table register (QuadSPI0_LUT3)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4320	Look-up Table register (QuadSPI0_LUT4)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4324	Look-up Table register (QuadSPI0_LUT5)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4328	Look-up Table register (QuadSPI0_LUT6)	32	R/W	0000_0000h	30.32.32/ 1240
4004_432C	Look-up Table register (QuadSPI0_LUT7)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4330	Look-up Table register (QuadSPI0_LUT8)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4334	Look-up Table register (QuadSPI0_LUT9)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4338	Look-up Table register (QuadSPI0_LUT10)	32	R/W	0000_0000h	30.32.32/ 1240
4004_433C	Look-up Table register (QuadSPI0_LUT11)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4340	Look-up Table register (QuadSPI0_LUT12)	32	R/W	0000_0000h	30.32.32/ 1240

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_4344	Look-up Table register (QuadSPI0_LUT13)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4348	Look-up Table register (QuadSPI0_LUT14)	32	R/W	0000_0000h	30.32.32/ 1240
4004_434C	Look-up Table register (QuadSPI0_LUT15)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4350	Look-up Table register (QuadSPI0_LUT16)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4354	Look-up Table register (QuadSPI0_LUT17)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4358	Look-up Table register (QuadSPI0_LUT18)	32	R/W	0000_0000h	30.32.32/ 1240
4004_435C	Look-up Table register (QuadSPI0_LUT19)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4360	Look-up Table register (QuadSPI0_LUT20)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4364	Look-up Table register (QuadSPI0_LUT21)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4368	Look-up Table register (QuadSPI0_LUT22)	32	R/W	0000_0000h	30.32.32/ 1240
4004_436C	Look-up Table register (QuadSPI0_LUT23)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4370	Look-up Table register (QuadSPI0_LUT24)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4374	Look-up Table register (QuadSPI0_LUT25)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4378	Look-up Table register (QuadSPI0_LUT26)	32	R/W	0000_0000h	30.32.32/ 1240
4004_437C	Look-up Table register (QuadSPI0_LUT27)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4380	Look-up Table register (QuadSPI0_LUT28)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4384	Look-up Table register (QuadSPI0_LUT29)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4388	Look-up Table register (QuadSPI0_LUT30)	32	R/W	0000_0000h	30.32.32/ 1240
4004_438C	Look-up Table register (QuadSPI0_LUT31)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4390	Look-up Table register (QuadSPI0_LUT32)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4394	Look-up Table register (QuadSPI0_LUT33)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4398	Look-up Table register (QuadSPI0_LUT34)	32	R/W	0000_0000h	30.32.32/ 1240

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_439C	Look-up Table register (QuadSPI0_LUT35)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43A0	Look-up Table register (QuadSPI0_LUT36)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43A4	Look-up Table register (QuadSPI0_LUT37)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43A8	Look-up Table register (QuadSPI0_LUT38)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43AC	Look-up Table register (QuadSPI0_LUT39)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43B0	Look-up Table register (QuadSPI0_LUT40)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43B4	Look-up Table register (QuadSPI0_LUT41)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43B8	Look-up Table register (QuadSPI0_LUT42)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43BC	Look-up Table register (QuadSPI0_LUT43)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43C0	Look-up Table register (QuadSPI0_LUT44)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43C4	Look-up Table register (QuadSPI0_LUT45)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43C8	Look-up Table register (QuadSPI0_LUT46)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43CC	Look-up Table register (QuadSPI0_LUT47)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43D0	Look-up Table register (QuadSPI0_LUT48)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43D4	Look-up Table register (QuadSPI0_LUT49)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43D8	Look-up Table register (QuadSPI0_LUT50)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43DC	Look-up Table register (QuadSPI0_LUT51)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43E0	Look-up Table register (QuadSPI0_LUT52)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43E4	Look-up Table register (QuadSPI0_LUT53)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43E8	Look-up Table register (QuadSPI0_LUT54)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43EC	Look-up Table register (QuadSPI0_LUT55)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43F0	Look-up Table register (QuadSPI0_LUT56)	32	R/W	0000_0000h	30.32.32/ 1240

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_43F4	Look-up Table register (QuadSPI0_LUT57)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43F8	Look-up Table register (QuadSPI0_LUT58)	32	R/W	0000_0000h	30.32.32/ 1240
4004_43FC	Look-up Table register (QuadSPI0_LUT59)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4400	Look-up Table register (QuadSPI0_LUT60)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4404	Look-up Table register (QuadSPI0_LUT61)	32	R/W	0000_0000h	30.32.32/ 1240
4004_4408	Look-up Table register (QuadSPI0_LUT62)	32	R/W	0000_0000h	30.32.32/ 1240
4004_440C	Look-up Table register (QuadSPI0_LUT63)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4000	Module Configuration Register (QuadSPI1_MCR)	32	R/W	000F_4000h	30.32.1/ 1207
400C_4008	IP Configuration Register (QuadSPI1_IPCR)	32	R/W	0000_0000h	30.32.2/ 1208
400C_400C	Flash Configuration Register (QuadSPI1_FLSHCR)	32	R/W	0000_0303h	30.32.3/ 1209
400C_4010	Buffer0 Configuration Register (QuadSPI1_BUF0CR)	32	R/W	0000_0000h	30.32.4/ 1210
400C_4014	Buffer1 Configuration Register (QuadSPI1_BUF1CR)	32	R/W	0000_0000h	30.32.5/ 1211
400C_4018	Buffer2 Configuration Register (QuadSPI1_BUF2CR)	32	R/W	0000_0000h	30.32.6/ 1211
400C_401C	Buffer3 Configuration Register (QuadSPI1_BUF3CR)	32	R/W	See section	30.32.7/ 1212
400C_4020	Buffer Generic Configuration Register (QuadSPI1_BFGENCR)	32	R/W	0000_0000h	30.32.8/ 1213
400C_4030	Buffer0 Top Index Register (QuadSPI1_BUF0IND)	32	R/W	0000_0000h	30.32.9/ 1214
400C_4034	Buffer1 Top Index Register (QuadSPI1_BUF1IND)	32	R/W	0000_0000h	30.32.10/ 1215
400C_4038	Buffer2 Top Index Register (QuadSPI1_BUF2IND)	32	R/W	0000_0000h	30.32.11/ 1215
400C_4100	Serial Flash Address Register (QuadSPI1_SFAR)	32	R/W	0000_0000h	30.32.12/ 1216
400C_4108	Sampling Register (QuadSPI1_SMPR)	32	R/W	0000_0000h	30.32.13/ 1217
400C_410C	RX Buffer Status Register (QuadSPI1_RBSR)	32	R	0000_0000h	30.32.14/ 1218
400C_4110	RX Buffer Control Register (QuadSPI1_RBCT)	32	R/W	0000_0000h	30.32.15/ 1219

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_4150	TX Buffer Status Register (QuadSPI1_TBSR)	32	R	0000_0000h	30.32.16/ 1220
400C_4154	TX Buffer Data Register (QuadSPI1_TBDR)	32	R/W	0000_0000h	30.32.17/ 1221
400C_415C	Status Register (QuadSPI1_SR)	32	R	0000_3800h	30.32.18/ 1222
400C_4160	Flag Register (QuadSPI1_FR)	32	w1c	0800_0000h	30.32.19/ 1225
400C_4164	Interrupt and DMA Request Select and Enable Register (QuadSPI1_RSER)	32	R/W	0000_0000h	30.32.20/ 1228
400C_4168	Sequence Suspend Status Register (QuadSPI1_SPNDST)	32	R	0000_0000h	30.32.21/ 1231
400C_416C	Sequence Pointer Clear Register (QuadSPI1_SPTRCLR)	32	R/W	0000_0000h	30.32.22/ 1233
400C_4180	Serial Flash A1 Top Address (QuadSPI1_SFA1AD)	32	R/W	0000_0000h	30.32.23/ 1233
400C_4184	Serial Flash A2 Top Address (QuadSPI1_SFA2AD)	32	R/W	0000_0000h	30.32.24/ 1234
400C_4188	Serial Flash B1Top Address (QuadSPI1_SFB1AD)	32	R/W	0000_0000h	30.32.25/ 1234
400C_418C	Serial Flash B2Top Address (QuadSPI1_SFB2AD)	32	R/W	0000_0000h	30.32.26/ 1235
400C_4200	RX Buffer Data Register (QuadSPI1_RBDR0)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4204	RX Buffer Data Register (QuadSPI1_RBDR1)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4208	RX Buffer Data Register (QuadSPI1_RBDR2)	32	R/W	0000_0000h	30.32.27/ 1235
400C_420C	RX Buffer Data Register (QuadSPI1_RBDR3)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4210	RX Buffer Data Register (QuadSPI1_RBDR4)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4214	RX Buffer Data Register (QuadSPI1_RBDR5)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4218	RX Buffer Data Register (QuadSPI1_RBDR6)	32	R/W	0000_0000h	30.32.27/ 1235
400C_421C	RX Buffer Data Register (QuadSPI1_RBDR7)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4220	RX Buffer Data Register (QuadSPI1_RBDR8)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4224	RX Buffer Data Register (QuadSPI1_RBDR9)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4228	RX Buffer Data Register (QuadSPI1_RBDR10)	32	R/W	0000_0000h	30.32.27/ 1235

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_422C	RX Buffer Data Register (QuadSPI1_RBDR11)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4230	RX Buffer Data Register (QuadSPI1_RBDR12)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4234	RX Buffer Data Register (QuadSPI1_RBDR13)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4238	RX Buffer Data Register (QuadSPI1_RBDR14)	32	R/W	0000_0000h	30.32.27/ 1235
400C_423C	RX Buffer Data Register (QuadSPI1_RBDR15)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4240	RX Buffer Data Register (QuadSPI1_RBDR16)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4244	RX Buffer Data Register (QuadSPI1_RBDR17)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4248	RX Buffer Data Register (QuadSPI1_RBDR18)	32	R/W	0000_0000h	30.32.27/ 1235
400C_424C	RX Buffer Data Register (QuadSPI1_RBDR19)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4250	RX Buffer Data Register (QuadSPI1_RBDR20)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4254	RX Buffer Data Register (QuadSPI1_RBDR21)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4258	RX Buffer Data Register (QuadSPI1_RBDR22)	32	R/W	0000_0000h	30.32.27/ 1235
400C_425C	RX Buffer Data Register (QuadSPI1_RBDR23)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4260	RX Buffer Data Register (QuadSPI1_RBDR24)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4264	RX Buffer Data Register (QuadSPI1_RBDR25)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4268	RX Buffer Data Register (QuadSPI1_RBDR26)	32	R/W	0000_0000h	30.32.27/ 1235
400C_426C	RX Buffer Data Register (QuadSPI1_RBDR27)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4270	RX Buffer Data Register (QuadSPI1_RBDR28)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4274	RX Buffer Data Register (QuadSPI1_RBDR29)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4278	RX Buffer Data Register (QuadSPI1_RBDR30)	32	R/W	0000_0000h	30.32.27/ 1235
400C_427C	RX Buffer Data Register (QuadSPI1_RBDR31)	32	R/W	0000_0000h	30.32.27/ 1235
400C_4300	LUT Key Register (QuadSPI1_LUTKEY)	32	R/W	5AF0_5AF0h	30.32.28/ 1236

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_4304	LUT Lock Configuration Register (QuadSPI1_LCKCR)	32	R/W	0000_0002h	30.32.29/ 1237
400C_4310	Look-up Table register (QuadSPI1_LUT0)	32	R/W	0818_0403h	30.32.30/ 1238
400C_4314	Look-up Table register (QuadSPI1_LUT1)	32	R/W	2400_1C08h	30.32.31/ 1239
400C_4318	Look-up Table register (QuadSPI1_LUT2)	32	R/W	0000_0000h	30.32.32/ 1240
400C_431C	Look-up Table register (QuadSPI1_LUT3)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4320	Look-up Table register (QuadSPI1_LUT4)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4324	Look-up Table register (QuadSPI1_LUT5)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4328	Look-up Table register (QuadSPI1_LUT6)	32	R/W	0000_0000h	30.32.32/ 1240
400C_432C	Look-up Table register (QuadSPI1_LUT7)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4330	Look-up Table register (QuadSPI1_LUT8)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4334	Look-up Table register (QuadSPI1_LUT9)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4338	Look-up Table register (QuadSPI1_LUT10)	32	R/W	0000_0000h	30.32.32/ 1240
400C_433C	Look-up Table register (QuadSPI1_LUT11)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4340	Look-up Table register (QuadSPI1_LUT12)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4344	Look-up Table register (QuadSPI1_LUT13)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4348	Look-up Table register (QuadSPI1_LUT14)	32	R/W	0000_0000h	30.32.32/ 1240
400C_434C	Look-up Table register (QuadSPI1_LUT15)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4350	Look-up Table register (QuadSPI1_LUT16)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4354	Look-up Table register (QuadSPI1_LUT17)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4358	Look-up Table register (QuadSPI1_LUT18)	32	R/W	0000_0000h	30.32.32/ 1240
400C_435C	Look-up Table register (QuadSPI1_LUT19)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4360	Look-up Table register (QuadSPI1_LUT20)	32	R/W	0000_0000h	30.32.32/ 1240

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_4364	Look-up Table register (QuadSPI1_LUT21)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4368	Look-up Table register (QuadSPI1_LUT22)	32	R/W	0000_0000h	30.32.32/ 1240
400C_436C	Look-up Table register (QuadSPI1_LUT23)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4370	Look-up Table register (QuadSPI1_LUT24)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4374	Look-up Table register (QuadSPI1_LUT25)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4378	Look-up Table register (QuadSPI1_LUT26)	32	R/W	0000_0000h	30.32.32/ 1240
400C_437C	Look-up Table register (QuadSPI1_LUT27)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4380	Look-up Table register (QuadSPI1_LUT28)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4384	Look-up Table register (QuadSPI1_LUT29)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4388	Look-up Table register (QuadSPI1_LUT30)	32	R/W	0000_0000h	30.32.32/ 1240
400C_438C	Look-up Table register (QuadSPI1_LUT31)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4390	Look-up Table register (QuadSPI1_LUT32)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4394	Look-up Table register (QuadSPI1_LUT33)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4398	Look-up Table register (QuadSPI1_LUT34)	32	R/W	0000_0000h	30.32.32/ 1240
400C_439C	Look-up Table register (QuadSPI1_LUT35)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43A0	Look-up Table register (QuadSPI1_LUT36)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43A4	Look-up Table register (QuadSPI1_LUT37)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43A8	Look-up Table register (QuadSPI1_LUT38)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43AC	Look-up Table register (QuadSPI1_LUT39)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43B0	Look-up Table register (QuadSPI1_LUT40)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43B4	Look-up Table register (QuadSPI1_LUT41)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43B8	Look-up Table register (QuadSPI1_LUT42)	32	R/W	0000_0000h	30.32.32/ 1240

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_43BC	Look-up Table register (QuadSPI1_LUT43)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43C0	Look-up Table register (QuadSPI1_LUT44)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43C4	Look-up Table register (QuadSPI1_LUT45)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43C8	Look-up Table register (QuadSPI1_LUT46)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43CC	Look-up Table register (QuadSPI1_LUT47)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43D0	Look-up Table register (QuadSPI1_LUT48)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43D4	Look-up Table register (QuadSPI1_LUT49)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43D8	Look-up Table register (QuadSPI1_LUT50)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43DC	Look-up Table register (QuadSPI1_LUT51)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43E0	Look-up Table register (QuadSPI1_LUT52)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43E4	Look-up Table register (QuadSPI1_LUT53)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43E8	Look-up Table register (QuadSPI1_LUT54)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43EC	Look-up Table register (QuadSPI1_LUT55)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43F0	Look-up Table register (QuadSPI1_LUT56)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43F4	Look-up Table register (QuadSPI1_LUT57)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43F8	Look-up Table register (QuadSPI1_LUT58)	32	R/W	0000_0000h	30.32.32/ 1240
400C_43FC	Look-up Table register (QuadSPI1_LUT59)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4400	Look-up Table register (QuadSPI1_LUT60)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4404	Look-up Table register (QuadSPI1_LUT61)	32	R/W	0000_0000h	30.32.32/ 1240
400C_4408	Look-up Table register (QuadSPI1_LUT62)	32	R/W	0000_0000h	30.32.32/ 1240
400C_440C	Look-up Table register (QuadSPI1_LUT63)	32	R/W	0000_0000h	30.32.32/ 1240

30.32.1 Module Configuration Register (QuadSPIx_MCR)

The QuadSPI_MCR holds configuration data associated with QuadSPI operation.

Write:

- *All other fields: Anytime*

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								Reserved				Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	MDIS	Reserved		CLR_TXF	CLR_RXF	Reserved		DDR_EN	DQS_EN	Reserved		END_CFG		SWRSTHD	SWRSTSD
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_MCR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This field is reserved and should always be set to 0.
23–20 Reserved	This field is reserved.
19–16 Reserved	This field is reserved. This field is reserved and should always be set to 0xF.
15 Reserved	This field is reserved. This field is reserved.
14 MDIS	Module Disable. The MDIS bit allows the clock to the non-memory mapped logic in the QuadSPI to be stopped, putting the QuadSPI in a software controlled power-saving state. Please refer to , for more information. 0 Enable QuadSPI clocks. 1 Allow external logic to disable QuadSPI clocks.
13–12 Reserved	This field is reserved.
11 CLR_TXF	Clear TX FIFO/Buffer. Invalidate the TX Buffer content. 0 No action. 1 Read and write pointers of the TX Buffer are reset to 0. QSPI_TBSR[TRCTR] is reset to 0.
10 CLR_RXF	Clear RX FIFO. Invalidate the RX Buffer.

Table continues on the next page...

QuadSPIx_MCR field descriptions (continued)

Field	Description
	0 No action. 1 Read and write pointers of the RX Buffer are reset to 0. QSPI_RBSR[RDBFL] is reset to 0.
9–8 Reserved	This field is reserved.
7 DDR_EN	DDR mode enable: 0 2x and 4x clocks are disabled for SDR instructions only 1 2x and 4x clocks are enabled supports both SDR and DDR instruction.
6 DQS_EN	DQS enable: This field is valid only when the DDR_EN field is set.For more details Refer Data Strobe Signal Functionality 0 DQS disabled. 1 DQS enabled- When enabled, the incoming data is sampled on both the edges of DQS input. The QSPI_SMPR[DDR_SMP] values are ignored.
5–4 Reserved	This field is reserved.
3–2 END_CFG	Defines the endianness of the QSPI module.For more details refer to Byte Ordering Endianess
1 SWRSTHD	Software reset for AHB domain 0 No action 1 AHB domain flops are reset. Does not reset configuration registers. It is advisable to reset both the serial flash domain and AHB domain at the same time. Resetting only one domain might lead to side effects.
0 SWRSTSD	Software reset for Serial Flash domain 0 No action 1 Serial Flash domain flops are reset. Does not reset configuration registers. It is advisable to reset both the serial flash domain and AHB domain at the same time. Resetting only one domain might lead to side effects.

30.32.2 IP Configuration Register (QuadSPIx_IPCR)

The IP configuration register provides all the configuration required for an IP initiated command. An IP command can be triggered by writing in the SEQID field of this register. If the SEQID field is written successfully, a new command to the external serial flash is started as per the sequence pointed to by the SEQID field. Refer to Normal Mode , for details about the command triggering and command execution.

Write:

- *QSPI_SR[IP_ACC]=0*

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved				SEQID				Reserved							PAR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	IDATSZ															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_IPCR field descriptions

Field	Description
31–28 Reserved	This field is reserved.
27–24 SEQID	Points to a sequence in the Look-up-table. The SEQID defines the bits [6:2] of the LUT index. The bits [1:0] are always assumed to be 0. Refer to Look-up Table for more details. A write to this bit -field triggers a transaction on the serial flash interface.
23–17 Reserved	This field is reserved.
16 PAR_EN	When set, a transaction to two serial flash devices is triggered in parallel mode. Refer to Parallel Flash Mode for more details.
15–0 IDATSZ	IP data transfer size: Defines the data transfer size in bytes of the IP command.

30.32.3 Flash Configuration Register (QuadSPIx_FLSHCR)

The Flash configuration register contains the flash device specific timings that must be met by the QuadSPI controller for the device to function correctly.

Write:

- *QSPI_SR[AHB_ACC] = 0*
- *QSPI_SR[IP_ACC] = 0*

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved																				TCSH				Reserved				TCSS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

QuadSPIx_FLSHCR field descriptions

Field	Description
31–12 Reserved	This field is reserved. Reserved.
11–8 TCSH	Serial flash CS hold time in terms of serial flash clock cycles.
7–4 Reserved	This field is reserved. Reserved.
3–0 TCSS	Serial flash CS setup time in terms of serial flash clock cycles.

30.32.4 Buffer0 Configuration Register (QuadSPIx_BUF0CR)

This register provides the configuration for any access to buffer0. An access is routed to buffer0 when the master port number of the incoming AHB request matches the MSTRID field of the BUF0CR. Any buffer "miss" leads to a serial flash transaction being triggered as per the sequence pointed to the SEQID field. Buffer0 may also be configured as a high priority buffer by setting the HP_EN field of this register.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HP_EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADATSZ								Reserved				MSTRID			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_BUF0CR field descriptions

Field	Description
31 HP_EN	High Priority Enable: When set, the master associated with this buffer is assigned a priority higher than the rest of the masters. An access by a high priority master will suspend any ongoing prefetch by another AHB master and will be serviced on high priority. Refer to Flexible AHB Buffers for details.
30–16 Reserved	This field is reserved.
15–8 ADATSZ	AHB data transfer size: Defines the data transfer size in 8 bytes of an AHB triggered access to serial flash. For example,a value of 0x2 will set transfer size to 16bytes.When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. SW should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved. Reserved.

Table continues on the next page...

QuadSPIx_BUF0CR field descriptions (continued)

Field	Description
3–0 MSTRID	Master ID: The ID of the AHB master associated with BUFFER0. Any AHB access with this master port number is routed to this buffer.

30.32.5 Buffer1 Configuration Register (QuadSPIx_BUF1CR)

This register provides the configuration for any access to buffer1. An access is routed to buffer1 when the master port number of the incoming AHB request matches the MSTRID field of the BUF1CR. Any buffer "miss" leads to the buffer being flushed and a serial flash transaction being triggered as per the sequence pointed to by the SEQID field.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ADATSZ								Reserved				MSTRID			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_BUF1CR field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–8 ADATSZ	AHB data transfer size: Defines the data transfer size in 8 bytes of an AHB triggered access to serial flash.For example, a value of 0x2 will set transfer size to 16bytes. When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. SW should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved.
3–0 MSTRID	Master ID: The ID of the AHB master associated with BUFFER1. Any AHB access with this master port number is routed to this buffer.

30.32.6 Buffer2 Configuration Register (QuadSPIx_BUF2CR)

This register provides the configuration for any access to buffer2. An access is routed to buffer2 when the master port number of the incoming AHB request matches the MSTRID field of the BUF2CR. Any buffer "miss" leads to the buffer being flushed and a serial flash transaction being triggered as per the sequence pointed to by the SEQID field.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ADATSZ								Reserved				MSTRID			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_BUF2CR field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved.
15–8 ADATSZ	AHB data transfer size: Defines the data transfer size in 8 Bytes of an AHB triggered access to serial flash. For example, a value of 0x2 will set transfer size to 16bytes.When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. SW should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved. Reserved.
3–0 MSTRID	Master ID: The ID of the AHB master associated with BUFFER2. Any AHB access with this master port number is routed to this buffer.

30.32.7 Buffer3 Configuration Register (QuadSPIx_BUF3CR)

This register provides the configuration for any access to buffer3. An access is routed to buffer3 when the master port number of the incoming AHB request matches the MSTRID field of the BUF3CR. Any buffer "miss" leads to the buffer being flushed a serial flash transaction being triggered as per the sequence pointed to by the SEQID field. If the ALLMST field is set, any transaction where the master port number does not match any of the buffer MSTRID fields will be routed to this buffer. In the case that the ALLMST field is not set, any such transaction (where master port number does not match any of the MSTRID fields) will be returned an ERROR response.

Write:

- *QSPI_SR[AHB_ACC] = 0*

.

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ALLMST	Reserved														
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ADATSZ								Reserved				MSTRID			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_BUF3CR field descriptions

Field	Description
31 ALLMST	All master enable: When set, buffer3 acts as an all-master buffer. Any AHB access with a master port number not matching with the master ID of buffer0 or buffer1 or buffer2 is routed to buffer3. When set, the MSTRID field of this register is ignored.
30–16 Reserved	This field is reserved. Reserved.
15–8 ADATSZ	AHB data transfer size: Defines the data transfer size in 8 Bytes of an AHB triggered access to serial flash. When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. SW should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved.
3–0 MSTRID	Master ID: The ID of the AHB master associated with BUFFER3. Any AHB access with this master port number is routed to this buffer.

30.32.8 Buffer Generic Configuration Register (QuadSPIx_BFGENCR)

This register provides the generic configuration to any of the buffer accesses. Any buffer "miss" leads to the buffer being flushed and a serial flash transaction being triggered as per the sequence pointed to by the SEQID field. If the PAR_EN field is set, all the buffer accesses result in parallel accesses to the flashes.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															PAR_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	SEQID				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_BFGENCR field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 PAR_EN	When set, a transaction to two serial flash devices is triggered in parallel mode. Refer to Parallel Flash Mode for more details.
15–12 SEQID	Points to a sequence in the Look-up-table. The SEQID defines the bits [6:2] of the LUT index. The bits [1:0] are always assumed to be 0. Refer to Look-up Table. **NOTE:** If the sequence pointer differs between the new and previous sequence then the user should reset this. See QSPI_SPTRCLR for more information.
11–0 Reserved	This field is reserved.

30.32.9 Buffer0 Top Index Register (QuadSPIx_BUF0IND)

This register specifies the top index of buffer0, which defines its size. Note that that the 3 LSBs of this register are set to zero - this ensures that the buffer is 64bit aligned, as each buffer entry is 64bits long.

The register value should be set to the desired number of bytes less 8. For example, setting BUF0IND to 0 gives 8 bytes, 1 give 16bytes etc.

The size of buffer0 is the difference between the BUF0IND+8 and 0.

It is the responsibility of the software to ensure that BUF0IND value is not greater than the overall size of the buffer. The hardware does not provide any protection against illegal programming.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	TPINDX0																													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_BUF0IND field descriptions

Field	Description
31–3 TPINDX0	Top index of buffer 0.
2–0 Reserved	This field is reserved. Reserved.

30.32.10 Buffer1 Top Index Register (QuadSPIx_BUF1IND)

This register specifies the top index of buffer1, which defines its size. Note that the 3 LSBs of this register are set to zero - this ensures that the buffer is 64bit aligned as each buffer entry is 64bits long.

The register value should be set to the desired number of bytes less. The size of buffer1 is the difference between the BUF1IND and BUF0IND.

It is the responsibility of the software to ensure that BUF1IND value is not greater than the overall size of the buffer. The hardware does not provide any protection against illegal programming.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPINDX1																													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_BUF1IND field descriptions

Field	Description
31–3 TPINDX1	Top index of buffer 1.
2–0 Reserved	This field is reserved.

30.32.11 Buffer2 Top Index Register (QuadSPIx_BUF2IND)

This register specifies the top index of buffer2, which defines its size. Note that that the 3 LSBs of this register are set to zero - this ensures that the buffer is 64bit aligned as each buffer entry is 64bits long.

The register value should be set to the desired number of bytes less 8. The size of buffer2 is the difference between the BUF2IND and BUF1IND.

It is the responsibility of the software to ensure that BUF2IND value is not greater than the overall size of the buffer. The hardware does not provide any protection against illegal programming.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	TPINDX2																													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_BUF2IND field descriptions

Field	Description
31–3 TPINDX2	Top index of buffer 2.
2–0 Reserved	This field is reserved.

30.32.12 Serial Flash Address Register (QuadSPIx_SFAR)

The module automatically translates this address on the memory map to the address on the flash itself. When operating in 24bit mode, only bits 23-0 are sent to the flash, in 32bit mode, bits 27-0 are used with bits 31-28 driven to 0. Refer to Table 30-386 for the mapping between the access mode and the QSPI_SFAR content and to Normal Mode for details about the command triggering and command execution. The software should ensure that the serial flash address provided in the QSPI_SFAR register lies in the valid flash address range as defined in Table 30-386.

Write:

- *QSPI_SR[IP_ACC] = 0*

Address: Base address + 100h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	SFADR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_SFAR field descriptions

Field	Description
31–0 SFADR	Serial Flash Address. The register content is used as byte address for all following IP Commands.

30.32.13 Sampling Register (QuadSPIx_SMPR)

The Sampling Register allows configuration of how the incoming data from the external serial flash devices are sampled in the QuadSPI module.

Write:Disabled Mode

Address: Base address + 108h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													DDRSMP		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									FSDLY	FSPHS	0		HSDLY	HSPHS	HSENA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_SMPR field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 DDRSMP	DDR Sampling point. Select the sampling point for incoming data when serial flash is executing a DDR instruction. Refer to Figure 30-423 , for details on the sampling points.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FSDLY	Full Speed Delay selection for SDR instructions. Select the delay with respect to the reference edge for the sample point valid for full speed commands: 0 One clock cycle delay 1 Two clock cycles delay. **NOTE:** This bit is ignored when using DDR instructions
5 FSPHS	Full Speed Phase selection for SDR instructions. Select the edge of the sampling clock valid for full speed commands:

Table continues on the next page...

QuadSPIx_SMPR field descriptions (continued)

Field	Description
	0 Select sampling at non-inverted clock 1 Select sampling at inverted clock. **NOTE:** This bit is ignored when using DDR instructions.
4–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 HSDLY	Half Speed Delay selection for SDR instructions. Only relevant when HSENA bit is set. Select the delay with respect to the reference edge for the sample point valid for half speed commands: 0 One clock cycle delay 1 Two clock cycle delay
1 HSPHS	Half Speed Phase selection for SDR instructions. Only relevant when HSENA bit is set. Select the delay with respect to the reference edge for the sample point valid for half speed commands: 0 Select sampling at non-inverted clock 1 Select sampling at inverted clock
0 HSENA	Half Speed serial flash clock Enable: This bit enables the divide by 2 of the clock to the external serial flash device for all commands,both SDR and DDR. Refer to Serial Flash Clock Frequency Limitations for details. 0 Disable divide by 2 of serial flash clock for half speed commands 1 Enable divide by 2 of serial flash clock for half speed commands

30.32.14 RX Buffer Status Register (QuadSPIx_RBSR)

This register contains information related to the receive data buffer.

Address: Base address + 10Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDCTR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		RDBFL						Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_RBSR field descriptions

Field	Description
31–16 RDCTR	Read Counter, indicates how many entries of 4 bytes have been removed from the RX Buffer.For example a value of 0x2 would indicate 8bytes have been removed It is incremented by the number (QSPI_RBCT[WMRK] + 1) on RX Buffer POP event. The RX Buffer can be popped using DMA or pop flag QSPI_FR[RBDF]. The QSPI_RSER[RBDDE] defines which pop has to be done. For further details please refer to AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31) and "Data Transfer from the QuadSPI Module Internal Buffers section in Flash Read section.
15–14 Reserved	This field is reserved.
13–8 RDBFL	RX Buffer Fill Level, indicates how many entries of 4 bytes are still available in the RX Buffer.For example a value of 0x2 would indicate 8bytes are available.
7–0 Reserved	This field is reserved.

30.32.15 RX Buffer Control Register (QuadSPIx_RBCT)

This register contains control data related to the receive data buffer.

Write:

- *QSPI_SR[IP_ACC] = 0*

Address: Base address + 110h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved							RXBRD	Reserved			WMRK				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_RBCT field descriptions

Field	Description
31–9 Reserved	This field is reserved.
8 RXBRD	RX Buffer Readout: This bit specifies the access scheme for the RX Buffer readout.

Table continues on the next page...

QuadSPIx_RBCT field descriptions (continued)

Field	Description
	0 RX Buffer content is read using the AHB Bus registers QSPI_ARDB0 to QSPI_ARDB31. For details, refer to Exclusive Access to Serial Flash for AHB Commands. 1 RX Buffer content is read using the IP Bus registers QSPI_RBDR0 to QSPI_RBDR31.
7–5 Reserved	This field is reserved.
4–0 WMRK	RX Buffer Watermark: This field determines when the readout action of the RX Buffer is triggered. When the number of valid entries in the RX Buffer is equal to or greater than the number given by (WMRK+1) the QSPI_SR[RXWE] flag is asserted.The value should be entered as the number of 4byte entries minus 1. For example a value of 0x0 would set the watermark to 4bytes, 1 to 8bytes, 2 to 12bytes etc. For details, refer to DMA Usage.

30.32.16 TX Buffer Status Register (QuadSPIx_TBSR)

This register contains information related to the transmit data buffer.

Address: Base address + 150h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRCTR																Reserved			TRBFL					Reserved							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_TBSR field descriptions

Field	Description
31–16 TRCTR	Transmit Counter. This field indicates how many entries of 4 bytes have been written into the TX Buffer by host accesses. It is reset to 0 when a 1 is written into the QSPI_MCR[CLR_TXF] bit. It is incremented on each write access to the QSPI_TBDR register when another word has been pushed onto the TX Buffer. When it is not cleared the TRCTR field wraps around to 0. Refer to TX Buffer Data Register (QuadSPI_TBDR) for details.
15–13 Reserved	This field is reserved.
12–8 TRBFL	TX Buffer Fill Level. The TRBFL field contains the number of entries of 4 bytes each available in the TX Buffer for the QuadSPI module to transmit to the serial flash device.
7–0 Reserved	This field is reserved.

30.32.17 TX Buffer Data Register (QuadSPIx_TBDR)

The QSPI_TBDR register provides access to the circular TX Buffer of depth 64 bytes. This buffer provides the data written into it as write data for the page programming commands to the serial flash device. Refer to Table 30-429 for the byte ordering scheme. A write transaction on the flash with data size of less than 32 bits will lead to the removal of one data entry from the TX buffer. The valid bits will be used and the rest of the bits will be discarded.

Write:

- *QSPI_SR[TXFULL] = 0*

32-bit write access required

Address: Base address + 154h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_TBDR field descriptions

Field	Description
31–0 TXDATA	TX Data On write access the data is written into the next available entry of the TX Buffer and the QPSI_TBSR[TRBFL] field is updated accordingly. On a read access, the last data written to the register is returned.

30.32.18 Status Register (QuadSPIx_SR)

The QSPI_SR register provides all available status information about SFM command execution and arbitration, the RX Buffer and TX Buffer and the AHB Buffer.

Address: Base address + 15Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLPSMP			Reserved	TXFULL	Reserved		TXNE	RXDMA	Reserved			RXFULL	Reserved		RXWE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	AHB3FUL	AHB2FUL	AHB1FUL	AHB0FUL	AHB3NE	AHB2NE	AHB1NE	AHB0NE	AHBTRN	AHBGNT	Reserved	Reserved	AHB_ACC	IP_ACC	BUSY
W																
Reset	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_SR field descriptions

Field	Description
31–29 DLPSMP	Data learning pattern sampling point: The sampling point found by the controller with the data learning pattern. • This is used for DDR only. • If the learning fails, this field will return garbage and DLPFF bit will be set.
28 Reserved	This field is reserved.
27 TXFULL	TX Buffer Full: Asserted when no more data can be stored.
26–25 Reserved	This field is reserved.
24 TXNE	TX Buffer Not Empty: Asserted when TX Buffer contains data.
23 RXDMA	RX Buffer DMA: Asserted when RX Buffer read out via DMA is active i.e DMA is requested or running.
22–20 Reserved	This field is reserved.

Table continues on the next page...

QuadSPIx_SR field descriptions (continued)

Field	Description
19 RXFULL	RX Buffer Full: Asserted when the RX Buffer is full, i.e. that QSPI_RBSR[RDBFL] field is equal to 32.
18–17 Reserved	This field is reserved.
16 RXWE	RX Buffer Watermark Exceeded: Asserted when the number of valid entries in the RX Buffer exceeds the number given in the QSPI_RBCT[WMRK] field.
15 Reserved	This field is reserved.
14 AHB3FUL	AHB 3 Buffer Full: Asserted when AHB 3 buffer is full.
13 AHB2FUL	AHB 2 Buffer Full: Asserted when AHB 2 buffer is full.
12 AHB1FUL	AHB 1 Buffer Full: Asserted when AHB 1 buffer is full.
11 AHB0FUL	AHB 0 Buffer Full: Asserted when AHB 0 buffer is full.
10 AHB3NE	AHB 3 Buffer Not Empty: Asserted when AHB 3 buffer contains data.
9 AHB2NE	AHB 2 Buffer Not Empty: Asserted when AHB 2 buffer contains data.
8 AHB1NE	AHB 1 Buffer Not Empty: Asserted when AHB 1 buffer contains data.
7 AHB0NE	AHB 0 Buffer Not Empty: Asserted when AHB 0 buffer contains data.
6 AHBTRN	AHB Access Transaction pending: Asserted when there is a pending request on the AHB interface. Refer to the AMBA specification for details.
5 AHBGNT	AHB Command priority Granted: Asserted when another module has been granted priority of AHB Commands against IP Commands. For details refer to Command Arbitration.
4 Reserved	This field is reserved.
3 RESERVED	This field is reserved.
2 AHB_ACC	AHB Access: Asserted when the transaction currently executed was initiated by AHB bus.
1 IP_ACC	IP Access: Asserted when transaction currently executed was initiated by IP bus.
0 BUSY	Module Busy: Asserted when module is currently busy handling a transaction to an external flash device.

30.32.19 Flag Register (QuadSPIx_FR)

The QSPI_FR register provides all available flags about SFM command execution and arbitration which may serve as source for the generation of interrupt service requests. Note that the error flags in this register do not relate directly to the execution of the transaction in the serial flash device itself but only to the behavior and conditions visible in the QuadSPI module.

Write:Enabled Mode

Address: Base address + 160h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLPFF	Reserved		Reserved	TBFF	TBUF	Reserved		ILLINE	Reserved					RBOF	RBDF
W	w1c				w1c	w1c			w1c						w1c	w1c
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ABSEF	AITEF	AIBSEF	ABOF	IUEF	Reserved			IPAEF	IPIEF	Reserved	IPGEF	Reserved			TFF
W	w1c	w1c	w1c	w1c	w1c				w1c	w1c		w1c				w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_FR field descriptions

Field	Description
31 DLPFF	Data Learning Pattern Failure Flag: Set when DATA_LEARN instruction was encountered in a sequence but no sampling point was found for the data learning pattern. The controller automatically starts sampling using the value in QSPI_SMPR[DDRSMP].
30–29 RESERVED	This field is reserved.
28 Reserved	This field is reserved.
27 TBFF	TX Buffer Fill Flag: Before writing to the TX buffer, this bit should be cleared. Then this bit has to be read back. If the bit is set, the TX Buffer can take more data. If the bit remains cleared, the TX buffer is full. Refer to Tx Buffer Operation for details.
26 TBUF	TX Buffer Underrun Flag: Set when the module tried to pull data although TX Buffer was empty. The IP Command leading to the TX Buffer underrun is continued (data sent to the serial flash device is undefined). The application must clear the TX Buffer in response to this event by writing a 1 into the QSPI_MCR[CLR_TXF] bit.
25–24 Reserved	This field is reserved.
23 ILLINE	Illegal Instruction Error Flag: Set when an illegal instruction is encountered by the controller in any of the sequences. Refer to Table 30-427 for a list of legal instructions.
22–18 Reserved	This field is reserved.
17 RBOF	RX Buffer Overflow Flag: Set when not all the data read from the serial flash device could be pushed into the RX Buffer. The IP Command leading to this condition is continued until the number of bytes according to the QSPI_IPCR[IDATSZ] field has been read from the serial flash device. The content of the RX Buffer is not changed.

Table continues on the next page...

QuadSPIx_FR field descriptions (continued)

Field	Description
16 RBDF	RX Buffer Drain Flag: Will be set if the QuadSPI_SR[RXWE] status bit is asserted. Writing 1 into this bit triggers one of the following actions: • If the RX Buffer has up to QuadSPI_RBCT[WMRK] valid entries then the flag is cleared. • If the RX Buffer has more than QuadSPI_RBCT[WMRK] valid entries and the QuadSPI_RSER[RBDDE] bit is not set (flag driven mode) a RX Buffer POP event is triggered. The flag remains set if the RX Buffer contains more than QuadSPI_RBCT[WMRK] valid entries after the RX Buffer POP event is finished. The flag is cleared if the RX Buffer contains less than or equal to QuadSPI_RBCT[WMRK] valid entries after the RX Buffer POP event is finished. Refer to "Receive Buffer Drain Interrupt or DMA Request" section in Normal Mode Interrupt and DMA Requests, for details.
15 ABSEF	AHB Sequence Error Flag: Set when the execution of an AHB Command is started with an WRITE or WRITE_DDR Command in the sequence pointed to by the QSPI_BUFxCR [1] register No communication with the serial flash device is initiated by the QuadSPI module. The AHB bus request which triggered this command is answered with an ERROR response.
14 AITEF	AHB Illegal transaction error flag.Set whenever there is no response generated from QSPI to AHB bus in case of illegal transaction and the watchdog timer expires.The timer value is taken as parameter.
13 AIBSEF	AHB Illegal Burst Size Error Flag: Set whenever the total burst size(size x beat) of an AHB transaction is greater than the prefetch data size. The prefecth data size is defined by QSPI_BUFxCR[ADATSZ] or data size mentioned in the sequence pointed to by the SEQID field in case ADATSZ =0. Refer to HBURST Support for more details on hburst feature.
12 ABOF	AHB Buffer Overflow Flag: Set when the size of the AHB access exceeds the size of the AHB buffer. This condition can occur only if the QSPI_BUFxCR[ADATSZ] field is programmed incorrectly. The AHB Command leading to this condition is continued until the number of entries according to the QSPI_BUFxCR[ADATSZ] field has been read from the serial flash device. The content of the AHB Buffer is not changed.
11 IUEF	IP Command Usage Error Flag: Set when in parallel flash mode the execution of an IP Command is started with more than one pad enabled and the sequence pointed to by the sequence ID contains a WRITE or a WRITE_DDR command. Refer to Table 30-427 table for the related commands. No communication with the serial flash device is initiated by the QuadSPI module.
10–8 Reserved	This field is reserved.
7 IPAEF	IP Command Trigger during AHB Access Error Flag. Set when the following condition occurs: • A write access occurs to the QSPI_IPCR[SEQID] field and the QSPI_SR[AHB_ACC] bit is set. Any command leading to the assertion of the IPAEF flag is ignored.
6 IPIEF	IP Command Trigger could not be executed Error Flag. Set when the QSPI_SR[IP_ACC] bit is set (i.e. an IP triggered command is currently executing) and any of the following conditions occurs: • Write access to the QSPI_IPCR register. Any command leading to the assertion of the IPIEF flag is ignored • Write access to the QSPI_SFAR register. • Write access to the QSPI_RBCT register.
5 Reserved	This field is reserved.
4 IPGEF	IP Command Trigger during AHB Grant Error Flag: Set when the following condition occurs: • A write access occurs to the QSPI_IPCR[SEQID] field and the QSPI_SR[AHBGNT] bit is set. Any command leading to the assertion of the IPGEF flag is ignored.

Table continues on the next page...

QuadSPIx_FR field descriptions (continued)

Field	Description
3–1 Reserved	This field is reserved.
0 TFF	IP Command Transaction Finished Flag: Set when the QuadSPI module has finished a running IP Command. If an error occurred the related error flags are valid, at the latest, in the same clock cycle when the TFF flag is asserted.

1. QSPI_BUFxCR implies anyone of QSPI_BUF0CR/QSPI_BUF1CR/QSPI_BUF2CR/QSPI_BUF3CR

30.32.20 Interrupt and DMA Request Select and Enable Register (QuadSPIx_RSER)

The QuadSPI_RSER register provides enables and selectors for the interrupts in the QuadSPI module.

NOTE

Each flag of the QuadSPI_FR register enabled as source for an interrupt prevents the QuadSPI module from entering Stop Mode or Module Disable Mode when this flag is set.

>

Write:Anytime

Address: Base address + 164h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	DLPFIE	Reserved		Reserved	TBFIE	TBUIE	Reserved		ILLINIE	Reserved	RBDDE	Reserved			RBOIE	RBDIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ABSEIE	AITIE	AIBSIE	ABOIE	IUEIE	Reserved			IPAEIE	IPIEIE	Reserved	IPGEIE	Reserved			TFIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_RSER field descriptions

Field	Description
31 DLPFIE	Data Learning Pattern Failure Interrupt enable . Triggered by DLPFF flag in QSPI_FR register
30–29 RESERVED	This field is reserved.
28 Reserved	This field is reserved.
27 TBFIE	TX Buffer Fill Interrupt Enable
26 TBUIE	TX Buffer Underrun Interrupt Enable
25–24 Reserved	This field is reserved.
23 ILLINIE	Illegal Instruction Error Interrupt Enable. Triggered by ILLINE flag in QSPI_FR
22 Reserved	This field is reserved.
21 RBDDE	RX Buffer Drain DMA Enable: Enables generation of DMA requests for RX Buffer Drain. When this bit is set DMA requests are generated as long as the QSPI_SR[RXWE] status bit is set. 0 No DMA request will be generated 1 DMA request will be generated
20–18 Reserved	This field is reserved.
17 RBOIE	RX Buffer Overflow Interrupt Enable
16 RBDIE	RX Buffer Drain Interrupt Enable: Enables generation of IRQ requests for RX Buffer Drain. When this bit is set the interrupt is asserted as long as the QuadSPI_SR[RBDF] flag is set. 0 No RBDF interrupt will be generated 1 RBDF Interrupt will be generated
15 ABSEIE	AHB Sequence Error Interrupt Enable: Triggered by ABSEF flags of QSPI_FR
14 AITIE	AHB Illegal transaction interrupt enable.
13 AIBSIE	AHB Illegal Burst Size Interrupt Enable
12 ABOIE	AHB Buffer Overflow Interrupt Enable
11 IUEIE	IP Command Usage Error Interrupt Enable
10–8 Reserved	This field is reserved.
7 IPAEIE	IP Command Trigger during AHB Access Error Interrupt Enable
6 IPIEIE	IP Command Trigger during IP Access Error Interrupt Enable

Table continues on the next page...

QuadSPIx_RSER field descriptions (continued)

Field	Description
5 Reserved	This field is reserved.
4 IPGEIE	IP Command Trigger during AHB Grant Error Interrupt Enable
3–1 Reserved	This field is reserved. Reserved.
0 TFIE	Transaction Finished Interrupt Enable

30.32.21 Sequence Suspend Status Register (QuadSPIx_SPNDST)

The sequence suspend status register provides information specific to any suspended sequence. An AHB sequence may be suspended when a high priority AHB master makes an access before the AHB sequence completes the data transfer requested.

Address: Base address + 168h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATLFT							Reserved	SPDBUF		Reserved					SUSPND
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_SPNDST field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–9 DATLFT	Data left: Provides information about the amount of data left to be read in the suspended sequence. Valid only when SUSPND is set to 1'b1. Value in terms of 64 bits or 8 bytes
8 Reserved	This field is reserved.
7–6 SPDBUF	Suspended Buffer: Provides the suspended buffer number. Valid only when SUSPND is set to 1'b1
5–1 Reserved	This field is reserved.
0 SUSPND	When set, it signifies that a sequence is in suspended state

30.32.22 Sequence Pointer Clear Register (QuadSPIx_SPTRCLR)

The sequence pointer clear register provides bits to reset the IP and Buffer sequence pointers. The sequence pointer contains the index of which instruction within the LUT entry is to be executed next. For example, if the LUT entry ends on a JMP_ON_CS value of 2, the index will be stored as 2.

The software should reset the sequence pointers whenever the sequence ID is changed by updating the SEQID field in QSPI_IPCR or QSPI_BFGENCR.

Address: Base address + 16Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved							IPPTRC	Reserved							BFPTRC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_SPTRCLR field descriptions

Field	Description
31–9 Reserved	This field is reserved.
8 IPPTRC	IP Pointer Clear: 1: Clears the sequence pointer for IP accesses as defined in QuadSPI_IPCR
7–1 Reserved	This field is reserved. Reserved.
0 BFPTRC	Buffer Pointer Clear: 1: Clears the sequence pointer for AHB accesses as defined in QuadSPI_BFGENCR.

30.32.23 Serial Flash A1 Top Address (QuadSPIx_SFA1AD)

The QSPI_SFA1AD register provides the address mapping for the serial flash A1.The difference between QSPI_SFA1AD[TPADA1] and QSPI_AMBA_BASE defines the size of the memory map for serial flash A1.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 180h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADA1																						Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_SFA1AD field descriptions

Field	Description
31–10 TPADA1	Top address for Serial Flash A1. In effect, TPADxx is the first location of the next memory.
9–0 Reserved	This field is reserved.

30.32.24 Serial Flash A2 Top Address (QuadSPIx_SFA2AD)

The QSPI_SFA2AD register provides the address mapping for the serial flash A2.The difference between QSPI_SFA2AD[TPADA2] and QSPI_SFA1AD[TPADA1] defines the size of the memory map for serial flash A2.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 184h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADA2																						Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_SFA2AD field descriptions

Field	Description
31–10 TPADA2	Top address for Serial Flash A2. In effect, TPxxAD is the first location of the next memory.
9–0 Reserved	This field is reserved.

30.32.25 Serial Flash B1Top Address (QuadSPIx_SFB1AD)

The QSPI_SFB1AD register provides the address mapping for the serial flash B1.The difference between QSPI_SFB1AD[TPADB1] and QSPI_SFA2AD[TPADA2] defines the size of the memory map for serial flash B1.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 188h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADB1																						Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_SFB1AD field descriptions

Field	Description
31–10 TPADB1	Top address for Serial Flash B1.In effect, TPxxAD is the first location of the next memory.
9–0 Reserved	This field is reserved.

30.32.26 Serial Flash B2Top Address (QuadSPIx_SFB2AD)

The QSPI_SFB2AD register provides the address mapping for the serial flash B2.The difference between QSPI_SFB2AD[TPADB2] and QSPI_SFB1AD[TPADB1] defines the size of the memory map for serial flash B2.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: Base address + 18Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADB2																						Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_SFB2AD field descriptions

Field	Description
31–10 TPADB2	Top address for Serial Flash B2. In effect, TPxxAD is the first location of the next memory.
9–0 Reserved	This field is reserved.

30.32.27 RX Buffer Data Register (QuadSPIx_RBDR*n*)

The QuadSPI_RBDR registers provide access to the individual entries in the RX Buffer. Refer to Table 30-429 for the byte ordering scheme.

QuadSPI_RBDR0 corresponds to the actual position of the read pointer within the RX Buffer. The number of valid entries available depends from the number of RX Buffer entries implemented and from the number of valid buffer entries available in the RX Buffer.

Example 1, RX Buffer filled completely with 32 words: In this case the address range for valid read access extends from QuadSPI_RBDR0 to QuadSPI_RBDR31.

Example 2, RX Buffer filled with 5 valid words: RX Buffer fill level QuadSPI_RBSR[RDBFL] is 5. In this case an access to QuadSPI_RBDR4 provides the last valid entry.

Any access beyond the range of valid RX Buffer entries provides undefined results.

Address: Base address + 200h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RXDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_RBDR*n* field descriptions

Field	Description
31–0 RXDATA	RX Data. The RXDATA field contains the data associated with the related RX Buffer entry. Data format and byte ordering is given in Byte Ordering of Serial Flash Read Data .

30.32.28 LUT Key Register (QuadSPIx_LUTKEY)

The LUT Key register contains the key to lock and unlock the Look-up-table. Refer to Look-up Table for details.

Write:Anytime

Address: Base address + 300h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	KEY																															
Reset	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	1	0	1	1	0	1	0	1	1	1	1	0	0	0	0

QuadSPIx_LUTKEY field descriptions

Field	Description
31–0 KEY	The key to lock or unlock the LUT. The KEY is 0x5AF05AF0. The read value is always 0x5AF05AF0

30.32.29 LUT Lock Configuration Register (QuadSPIx_LCKCR)

The LUT lock configuration register is used along with QSPI_LUTKEY register to lock or unlock the LUT. This register has to be written immediately after QSPI_LUTKEY register for the lock or unlock operation to be successful. Refer to Look-up Table for details. Setting both the LOCK and UNLOCK bits as "00" or "11" is not allowed.

Write:Just after writing the LUT Key Register

(QSPI_LUTKEY)

Address: Base address + 304h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved														UNLOCK	LOCK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

QuadSPIx_LCKCR field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1 UNLOCK	Unlocks the LUT when the following two conditions are met: 1. This register is written just after the LUT Key Register (QuadSPI_LUTKEY) 2. The LUT key register was written with 0x5AF05AF0 key
0 LOCK	Locks the LUT when the following condition is met: 1. This register is written just after the LUT Key Register (QuadSPI_LUTKEY) 2. The LUT key register was written with 0x5AF05AF0 key

30.32.30 Look-up Table register (QuadSPIx_LUT0)

The LUT registers are a look-up-table for sequences of instructions. The programmable sequence engine executes the instructions in these sequences to generate a valid serial flash transaction. There are a total of 64 LUT registers. These 64 registers are divided into groups of 4 registers that make a valid sequence. Therefore, QSPI_LUT[0], QSPI_LUT[4], QSPI_LUT[8] ..... QSPI_LUT[60] are the starting registers of a valid sequence. Each of these sets of 4 registers can have a maximum of 8 instructions. A maximum of 16 sequences can be defined at one time. Look-up Table describes the LUT registers in detail.

Write:Once the LUT is unlocked

Address: Base address + 310h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	INSTR1						PAD1		OPRND1							
Reset	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	INSTR0						PAD0		OPRND0							
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1

QuadSPIx_LUT0 field descriptions

Field	Description
31–26 INSTR1	Instruction 1
25–24 PAD1	Pad information for INSTR1. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
23–16 OPRND1	Operand for INSTR1.
15–10 INSTR0	Instruction 0
9–8 PAD0	Pad information for INSTR0. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
7–0 OPRND0	Operand for INSTR0.

30.32.31 Look-up Table register (QuadSPIx_LUT1)

The LUT registers are a look-up-table for sequences of instructions. The programmable sequence engine executes the instructions in these sequences to generate a valid serial flash transaction. There are a total of 64 LUT registers. These 64 registers are divided into groups of 4 registers that make a valid sequence. Therefore, QSPI_LUT[0], QSPI_LUT[4], QSPI_LUT[8] ..... QSPI_LUT[60] are the starting registers of a valid sequence. Each of these sets of 4 registers can have a maximum of 8 instructions. A maximum of 16 sequences can be defined at one time. Look-up Table describes the LUT registers in detail.

Write:Once the LUT is unlocked

Address: Base address + 314h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INSTR1						PAD1		OPRND1							
Reset	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INSTR0						PAD0		OPRND0							
Reset	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0

QuadSPIx_LUT1 field descriptions

Field	Description
31–26 INSTR1	Instruction 1
25–24 PAD1	Pad information for INSTR1. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
23–16 OPRND1	Operand for INSTR1.
15–10 INSTR0	Instruction 0
9–8 PAD0	Pad information for INSTR0. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
7–0 OPRND0	Operand for INSTR0.

30.32.32 Look-up Table register (QuadSPIx_LUT*n*)

The LUT registers are a look-up-table for sequences of instructions. The programmable sequence engine executes the instructions in these sequences to generate a valid serial flash transaction. There are a total of 64 LUT registers. These 64 registers are divided into groups of 4 registers that make a valid sequence. Therefore, QSPI_LUT[0], QSPI_LUT[4], QSPI_LUT[8] ..... QSPI_LUT[60] are the starting registers of a valid sequence. Each of these sets of 4 registers can have a maximum of 8 instructions. A maximum of 16 sequences can be defined at one time. Look-up Table describes the LUT registers in detail.

Write:Once the LUT is unlocked

Address: Base address + 318h offset + (4d × i), where i=0d to 61d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INSTR1						PAD1		OPRND1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INSTR0						PAD0		OPRND0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPIx_LUT*n* field descriptions

Field	Description
31–26 INSTR1	Instruction 1
25–24 PAD1	Pad information for INSTR1. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
23–16 OPRND1	Operand for INSTR1.
15–10 INSTR0	Instruction 0
9–8 PAD0	Pad information for INSTR0. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
7–0 OPRND0	Operand for INSTR0.

30.3.3 Serial Flash Address Assignment

The serial flash address assignment may be modified by writing into Serial Flash A1 Top Address (QuadSPI_SFA1AD) and Serial Flash A2 Top Address (QuadSPI_SFA2AD) for device A and into Serial Flash B1Top Address (QuadSPI_SFB1AD) and Serial Flash B2Top Address (QuadSPI_SFB2AD) for device B. The following table shows how different access modes are related to the address specified for the next SFM Command. Note that this address assignment is valid for both IP and AHB commands.

Table 30-386. Serial Flash Address Assignment

Parameter	Function	Access Mode
QSPI_AMBA_BASE ((31:10) - 22 bits)	QuadSPI AHB base address	
TOP_ADDR_MEMA1(TPADA1)	Top address for the external flash A1 (first device of the dual die flash A, or the first of the two independent flashes sharing the IOFA)	Any access to the address space between TOP_ADDR_MEMA1 and QSPI_AMBA_BASE will be routed to **Serial Flash A1**
TOP_ADDR_MEMA2(TPADA2)	Top address for the external flash A2 (second device of the dual die flash A, or the second of the two independent flashes sharing the IOFA).	Any access to the address space between TOP_ADDR_MEMA2 and TOP_ADDR_MEMA1 will be routed to **Serial Flash A2**
TOP_ADDR_MEMB1(TPADB1)	Top address for the external flash B1 (first device of the dual die flash B, or the first of the two independent flashes sharing the IOFB)	Any access to the address space between TOP_ADDR_MEMB1 and TOP_ADDR_MEMA2 will be routed to **Serial Flash B1**
TOP_ADDR_MEMB2(TPADB2)	Top address for the external flash B2 (second device of the dual die flash B or the second of the two independent flashes sharing the IOFB)	Any access to the address space between TOP_ADDR_MEMB2 and TOP_ADDR_MEMB1 will be routed to **Serial Flash A2**

30.3.4 AMBA Bus Register Memory Map

QSPI_AMBA_BASE defines the address to be used as start address of the serial flash device as defined by the system memory map..

Table 30-387. QuadSPI AMBA Bus Memory Map

Address	Register Name
Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A	

Table continues on the next page...

Table 30-387. QuadSPI AMBA Bus Memory Map (continued)

Address	Register Name
QSPI_AMBA_BASE to (TOP_ADDR_MEMA2 - 0x01)	Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A Refer to Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A for details and to Table 30-429 and Table 30-434 for information about the byte ordering.
Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B	
TOP_ADDR_MEMA2 to (TOP_ADDR_MEMB2 - 0x01)	Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B Refer to Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B for details and to Table 30-429 and Table 30-434 for information about the byte ordering.
Parallel Flash Mode	
QSPI_AMBA_BASE to (TOP_ADDR_MEMB2 - 0x01)	Parallel Flash Mode Refer to Parallel Flash Mode for details and to Table 30-433 and Table 30-434 for information about the byte ordering.
AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31)	
QSPI_ARDB_BASE to… (32 * 4 Byte) QSPI_ARDB_BASE + 0x0000_01FF	AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31) Refer to Table 30-429 and Table 30-431 for information about the byte ordering.

Note

Any read access to non-implemented addresses will provide undefined results.

In case single die flash devices, TOP_ADDR_MEMA2 and TOP_ADDR_MEMB2 should be initialized/programmed to TOP_ADDR_MEMA1 and TOP_ADDR_MEMB1 respectively- in effect, setting the size of these devices to 0. This would ensure that the complete memory map is assigned to only one flash device.

Parallel Flash Mode is valid only for commands related to data read and data write in single io mode from the serial flash. The first device of flash A has to be paired with the first device of flash B and the second device of flash A has to be paired with the second device of flash B in parallel mode. Parallel mode is selected via the QSPI_BFGENCR[PAR_EN] bit for all masters in AHB driven mode and via the QSPI_IPCR[PAR_EN] in IP driven mode. In parallel mode, the incoming address (SFAR address in case of IP initiated transactions and the incoming AHB address in case of AHB initiated transactions) is divided by 2 and sent to the two flashes connected in parallel.

Any IP Command other than data read and write (through one pad) in Parallel Flash Mode will result in the assertion of the QSPI_FR[IUEF] flag and any AHB Command other than data read in Parallel Flash Mode will result in the assertion of the QSPI_FR[ABSEF] flag.

In the Individual Flash Modes, the 3/4 address bytes (as programmed in the instruction/operand in the sequence) available for the flash address is determined by SFADR [23:0] or SFADR [31:0]as given in the table above.

In Parallel Flash Mode, both flashes are read with the same starting address of 3/4 (as programmed in the instruction/operand in the sequence) bytes in size. This address is derived from SFADR [24:1] or SFADR [31:1] as given in the table above. The LSB of the SFADR field is used to select the appropriate bits of both flash devices to combine the byte corresponding to the selected address.

30.3.5 AHB Bus Register Memory Map Descriptions

This chapter contains definitions of registers in the AMBA address space.

30.3.5.1 AHB Bus Access Considerations

It has to be noted that all logic in the QuadSPI module implementing the AHB Bus access is designed to read the content of an external serial flash device. Therefore the following restrictions apply to the QuadSPI module with respect to accesses to the AHB bus:

- Any write access is answered with the ERROR condition according to the AMBA AHB Specification. No write occurs.
- Any AHB Command resulting in the assertion of the QSPI_FR[ABSEF] flag is answered with the ERROR condition according to the AMBA_AHB specification. The resulting AHB Command is ignored.
- AHB Bus access types fully supported are NONSEQ and BUSY.
- AHB access type SEQ is treated in the same way like NONSEQ. Refer to the AMBA AHB Specification for further details.

30.3.5.2 Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A

Starting with address QSPI_AMBA_BASE the content of the first external serial flash devices is mapped into the address space of the device containing the QuadSPI module. Serial flash address byte address 0x0 corresponds to bus address QSPI_AMBA_BASE with increasing order. Assuming that a dual-die flash is connected on the first set of external pads, the address space is divided into two parts, one for each device of the dual die package. Refer to the following table for the address mapping. The byte ordering for 32 bit access is given in Table 30-429 and for 64 bit read access the byte ordering is given in Table 30-434.

Table 30-388. Memory Mapped Individual Flash Mode - Flash A Address Scheme

<table>
<tr><th>Memory Mapped Address 32 Bit Access</th><th>Memory Mapped Address 64 Bit Access</th><th>Serial Flash Byte Address</th><th>Flash Device</th></tr>
<tr><td>QSPI_AMBA_BASE + 0x00</td><td rowspan="2">QSPI_AMBA_BASE + 0x00</td><td>0x00_0000 to 0x00_0003</td><td rowspan="5">A1</td></tr>
<tr><td>QSPI_AMBA_BASE + 0x04</td><td>0x00_0004 to 0x00_0007</td></tr>
<tr><td>…</td><td>…</td><td>…</td></tr>
<tr><td>TOP_ADDR_MEMA1 - 0x08</td><td rowspan="2">TOP_ADDR_MEMA1 - 0x08</td><td>(TOP_ADDR_MEMA1- 0x08) to (TOP_ADDR_MEMA1 - 0x04 -0x01)</td></tr>
<tr><td>TOP_ADDR_MEMA1 - 0x04</td><td>(TOP_ADDR_MEMA1 - 0x04) to (TOP_ADDR_MEMA1 - 0x01)</td></tr>
<tr><td>TOP_ADDR_MEMA1 + 0x00</td><td rowspan="2">TOP_ADDR_MEMA1 + 0x00_0000</td><td>0x00_0000 to 0x00_0003</td><td rowspan="5">A2</td></tr>
<tr><td>TOP_ADDR_MEMA1 + 0x04</td><td>0x00_0004 to 0x00_0007</td></tr>
<tr><td>…..</td><td>…</td><td>…</td></tr>
<tr><td>TOP_ADDR_MEMA2 - 0x08</td><td rowspan="2">TOP_ADDR_MEMA2 - 0x08</td><td>(TOP_ADDR_MEMA2 - 0x08) to (TOP_ADDR_MEMA2 - 0x04 - 0x01)</td></tr>
<tr><td>TOP_ADDR_MEMA2 - 0x04</td><td>(TOP_ADDR_MEMA2 - 0x04) to (TOP_ADDR_MEMA2 - 0x01)</td></tr>
</table>

The available address range depends from the size of the external serial flash device. Any access beyond the size of the external serial flash provides undefined results.

For details concerning the read process refer to Flash Read.

30.3.5.3 Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B

Starting with address TOP_ADDR_MEMA2 the content of the first external serial flash devices is mapped into the address space of the device containing the QuadSPI module. Serial flash address byte address 0x0 corresponds to bus address TOP_ADDR_MEMA2 with increasing order. Assuming that a dual-die flash is connected on the first set of external pads, the address space is divided into two parts, one for each device of the dual die package. Refer the following table for the address mapping. The byte ordering for 32 bit access is given in Table 30-429 and for 64 bit read access the byte ordering is given in Table 30-434.

Table 30-389. Memory Mapped Individual Flash Mode - Flash B Address Scheme

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash Byte Address	Flash Device
TOP_ADDR_MEMA2 + 0x00	TOP_ADDR_MEMA2 + 0x00	0x00_0000 to 0x00_0003	B1
TOP_ADDR_MEMA2 + 0x04		0x00_0004 to 0x00_0007	
…	…	…	
TOP_ADDR_MEMB1 - 0x08	TOP_ADDR_MEMB1 - 0x08	(TOP_ADDR_MEMB1- TOP_ADDR_MEMA2 - 0x08) to (TOP_ADDR_MEMB1 - TOP_ADDR_MEMA2 - 0x04 -0x01)	
TOP_ADDR_MEMB1 - 0x04		(TOP_ADDR_MEMB1- TOP_ADDR_MEMA2 - 0x04) to (TOP_ADDR_MEMB1- TOP_ADDR_MEMA2- 0x01)	
TOP_ADDR_MEMB1 + 0x00	TOP_ADDR_MEMB1 + 0x00_0000	0x00_0000 to 0x00_0003	B2
TOP_ADDR_MEMB1 + 0x04		0x00_0004 to 0x00_0007	
…..	…	…	
TOP_ADDR_MEMB2 - 0x08	TOP_ADDR_MEMA2 - 0x08	(TOP_ADDR_MEMB2 - TOP_ADDR_MEMB1- 0x08) to (TOP_ADDR_MEMB2 - TOP_ADDR_MEMB1 - 0x04 - 0x01)	
TOP_ADDR_MEMB2 - 0x04		(TOP_ADDR_MEMB2- TOP_ADDR_MEMB1 - 0x04) to (TOP_ADDR_MEMB2 - TOP_ADDR_MEMB1 - 0x01)	

The available address range depends from the size of the external serial flash device. Any access beyond the size of the external serial flash provides undefined results.

For details concerning the read process refer to Flash Read.

30.3.5.4 Parallel Flash Mode

Any of the AHB flexible-buffers can be configured to work in parallel flash mode by programming the QSPI_BFGENCR[PAR_EN] bit to '1'. When parallel mode is set, Flash A1 is paired with Flash B1 and Flash A2 is paired with Flash B2. In parallel mode, software should ensure that the size of Flash A1(A2) is equal to the size of Flash B1(B2).

Reads from any even AHB bus address provides bits [7:4] of both serial flash devices and reads from any odd AHB bus address provides bits [3:0] of both flash devices. Refer to the following table for the address mapping. The byte ordering for 32 bit access is given in Table 30-431 and for 64 bit read access the byte ordering is given in Table 30-434.

Table 30-390. Memory Mapped Parallel Flash Mode Address Scheme

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash A Byte Address	Serial Flash B Byte Address
QSPI_AMBA_BASE + 0x0000_0000	QSPI_AMBA_BASE + 0x00 For details, please refer to Parallel mode and Dual Die Flashes.	0x00_0000 - 0x00_0001	0x00_0000 - 0x00_0001
QSPI_AMBA_BASE + 0x0000_0004		0x00_002 - 0x00_0003	0x00_0002 - 0x00_0003
QSPI_AMBA_BASE + 0x0000_0008	QSPI_AMBA_BASE + 0x08	0x00_0004 - 0x00_0005	0x00_0004 - 0x00_0005
QSPI_AMBA_BASE + 0x0000_000C		0x00_0006 - 0x00_0007	0x00_0006 - 0x00_0007
…	…	…	…
TOP_ADDR_MEMB2 - 0x08	TOP_ADDR_MEMB2 - 0x08	(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE - 0x08)/2	(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE - 0x08)/2
TOP_ADDR_MEMB2 - 0x04		(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE - 0x04)/2 +0x01	(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE- 0x04)/2 + 0x01

The available address range covers 27 address bits, corresponding to 128 MB per flash device. The usable space depends from the size of the external serial flash devices. Any access beyond the size of the external serial flash provides undefined results.

For details concerning the read process refer to Flash Read.

30.3.5.5 AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31)

30.3.5.5.1 AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31)

memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
2000	AHB RX Data Buffer register (ARDB0)	32	R/W	0000_0000h	30.3551.1/ 1247
2004	AHB RX Data Buffer register (ARDB 1)	32	R/W	0000_0000h	30.3551.1/ 1247
2008	AHB RX Data Buffer register (ARDB 2)	32	R/W	0000_0000h	30.3551.1/ 1247
200C	AHB RX Data Buffer register (ARDB 3)	32	R/W	0000_0000h	30.3551.1/ 1247
2010	AHB RX Data Buffer register (ARDB 4)	32	R/W	0000_0000h	30.3551.1/ 1247
2014	AHB RX Data Buffer register (ARDB 5)	32	R/W	0000_0000h	30.3551.1/ 1247
2018	AHB RX Data Buffer register (ARDB 6)	32	R/W	0000_0000h	30.3551.1/ 1247
201C	AHB RX Data Buffer register (ARDB 7)	32	R/W	0000_0000h	30.3551.1/ 1247
2020	AHB RX Data Buffer register (ARDB 8)	32	R/W	0000_0000h	30.3551.1/ 1247
2024	AHB RX Data Buffer register (ARDB 9)	32	R/W	0000_0000h	30.3551.1/ 1247
2028	AHB RX Data Buffer register (ARDB 10)	32	R/W	0000_0000h	30.3551.1/ 1247
202C	AHB RX Data Buffer register (ARDB 11)	32	R/W	0000_0000h	30.3551.1/ 1247
2030	AHB RX Data Buffer register (ARDB 12)	32	R/W	0000_0000h	30.3551.1/ 1247
2034	AHB RX Data Buffer register (ARDB 13)	32	R/W	0000_0000h	30.3551.1/ 1247
2038	AHB RX Data Buffer register (ARDB 14)	32	R/W	0000_0000h	30.3551.1/ 1247
203C	AHB RX Data Buffer register (ARDB 15)	32	R/W	0000_0000h	30.3551.1/ 1247
2040	AHB RX Data Buffer register (ARDB 16)	32	R/W	0000_0000h	30.3551.1/ 1247
2044	AHB RX Data Buffer register (ARDB 17)	32	R/W	0000_0000h	30.3551.1/ 1247
2048	AHB RX Data Buffer register (ARDB 18)	32	R/W	0000_0000h	30.3551.1/ 1247

Table continues on the next page...

memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
204C	AHB RX Data Buffer register (ARDB 19)	32	R/W	0000_0000h	30.3551.1/ 1247
2050	AHB RX Data Buffer register (ARDB 20)	32	R/W	0000_0000h	30.3551.1/ 1247
2054	AHB RX Data Buffer register (ARDB 21)	32	R/W	0000_0000h	30.3551.1/ 1247
2058	AHB RX Data Buffer register (ARDB 22)	32	R/W	0000_0000h	30.3551.1/ 1247
205C	AHB RX Data Buffer register (ARDB 23)	32	R/W	0000_0000h	30.3551.1/ 1247
2060	AHB RX Data Buffer register (ARDB 24)	32	R/W	0000_0000h	30.3551.1/ 1247
2064	AHB RX Data Buffer register (ARDB 25)	32	R/W	0000_0000h	30.3551.1/ 1247
2068	AHB RX Data Buffer register (ARDB 26)	32	R/W	0000_0000h	30.3551.1/ 1247
206C	AHB RX Data Buffer register (ARDB 27)	32	R/W	0000_0000h	30.3551.1/ 1247
2070	AHB RX Data Buffer register (ARDB 28)	32	R/W	0000_0000h	30.3551.1/ 1247
2074	AHB RX Data Buffer register (ARDB 29)	32	R/W	0000_0000h	30.3551.1/ 1247
2078	AHB RX Data Buffer register (ARDB 30)	32	R/W	0000_0000h	30.3551.1/ 1247
207C	AHB RX Data Buffer register (ARDB 31)	32	R/W	0000_0000h	30.3551.1/ 1247

30.3551.1 AHB RX Data Buffer register (ARDB*n*)

The AHB RX Data Buffer register 0 to 31 can be used to read the buffer content of the RX Buffer from successive addresses. QSPI_ARDB0 corresponds to the RX Buffer register entry corresponding to the current value of the read pointer with increasing order.

The increment of the read pointer depends from the access scheme (DMA or flag-driven). Refer to "Data Transfer from the QuadSPI Module Internal Buffers" section in Flash Read section, RX Buffer, data read via register interface and AHB read, for the description of successive accesses to the RX Buffer content. Refer also to Byte Ordering of Serial Flash Read Data for the byte ordering scheme.

Valid address range accessible in the QSPI_ARDBn range depends from the number of RX Buffer entries implemented and from the number of valid buffer entries available in the RX Buffer.

- Example 1, RX Buffer filled completely with 32 words: In this case the address range for valid read access extends from QSPI_ARDB0 to QSPI_ARDB31.
- Example 2, RX Buffer filled with 5 valid words, RX Buffer fill level QSPI_RBSR[RDBFL] is 5. In this case an access to QSPI_ARDB4 provides the last valid entry.

Address: 0h base + 2000h offset + (4d × i), where i=0d to 31d

Bit	31–0
R W	ARXD
Reset	

ARDB*n* field descriptions

Field	Description
31–0 ARXD	ARDB provided RX Buffer Data. Byte order (endianess) is identical to the RX Buffer Data Registers.

30.4 Interrupt Signals

The interrupt request lines of the QuadSPI module are mapped to the internal flags according to the following table.

Table 30-425. Assignment of Interrupt Request Lines

IRQ/DMA line	QSPI_FR Flag	Interrupt Description
ipi_int_tfff	TBFF	TX Buffer Fill
ipi_int_tcf	TFF	Peripheral Command Transaction Finished
ipi_int_rfdf	RBDF	RX Buffer Drain
ipi_int_overrun		Buffer Overflow/Underrun Error Logical OR from:
	RBOF	RX Buffer Overrun
	TBUF	TX Buffer Underrun
	ABOF	AHB Buffer Overflow
ipi_int_cerr		Serial Flash Command Error Logical OR from:
	IPAEF	Peripheral access while AHB busy Error
	IPIEF	Peripheral Command could not be triggered Error
	IPGEF	Peripheral access while AHB Grant Error
	IUEF	Peripheral Command Usage Error
	AITEF	AHB Illegal trasaction

Table continues on the next page...

Table 30-425. Assignment of Interrupt Request Lines (continued)

IRQ/DMA line	QSPI_FR Flag	Interrupt Description
ipi_int_ored	DLPFF, TBFF, TFF, ILLINE, RBDF, RBOF, TBUF, ABSEF, ABOF, IPAEF, IPIEF, IPGEF, IUEF, AITEF	Logical OR from all the QSPI_FR flags mentioned

30.5 Functional Description

This section provides the functional information of the QuadSPI module.

30.5.1 Serial Flash Access Schemes

The Quad Serial Peripheral Interface (QuadSPI) block acts as an interface to one single or two external serial flash devices, each with up to 4 bidirectional data lines. Depending from the serial flash devices attached to the QuadSPI module the following access schemes are possible:

Table 30-426. Access Schemes for Serial Flash Data Access

Access Scheme	One Flash Device on Port A	One Flash Device on Port B	Two identical Flash Devices connected on Port A and Port B
Individual Flash Mode: Access to Flash A	**Yes**	**N/a**	**Yes**
Individual Flash Mode: Access to Flash B	**N/a**	**Yes**	**Yes**
Parallel Flash Mode: Read from Flash A and Flash B	**N/a**	**N/a**	**Yes**

Note

If two flash devices are accessed in Parallel Flash Mode, they are accessed with identical control signals. Special alignment on per-flash basis is **not** possible. It is within the responsibility of the application to ensure that the identical signals are applicable to both flash devices.

In Parallel Flash Mode, both external serial flash devices appear logically as one single memory doubled in size with respect to one individual flash device.

If two different flash devices are attached, they can be operated only in Individual Flash Mode.

In the Parallel Flash Mode, only data read commands and write command (in x1 mode) are supported. Any other IP Command will result in an error condition signaled by the assertion of the QSPI_FR[IUEF] flag and any other AHB Command will result in the assertion of the QSPI_FR[ABSEF] flag.

In the Individual Flash Mode, all supported commands are available.

Unless explicitly noted, all the following descriptions relate to the Individual Flash Mode.

30.5.2 Modes of Operation

Refer to QuadSPI Modes of Operation for an overview over the possible operational modes of the QuadSPI block.

- Normal Mode can be used for write or read accesses to an external serial flash device.
  - Serial Flash Write: Data can be programmed into the flash via the IP interface only. Refer to Flash Programming for further details.
  - Serial Flash Read: Read the contents of the serial flash device. Two separate read channels are available via RX Buffer and AHB Buffer, see Flash Read.
- Stop Mode: The mode is used for power management. When a request is made to enter Stop Mode, the QuadSPI block acknowledges the request and completes the SFM Command in progress, then the system clocks to the QuadSPI block may be shut off
- Module Disable Mode: The mode is used for power management. The clock to the non-memory mapped logic in the QuadSPI can be stopped while in Module Disable Mode. The module enters the mode by setting QSPI_MCR[MDIS].

30.5.3 Normal Mode

This mode is used to allow communication with an external serial flash device. Compared to the standard SPI protocol, this communication method uses up to 4 bidirectional data lines operating at high data rates. The communication to the external serial flash device consists of an instruction code and optional address, mode, dummy

and data transfers. The flexible programmable core engine described below is immune to a wide variety of command/protocol differences in the serial flash devices provided by various flash vendors.

30.5.3.1 Programmable Sequence Engine

The core of the QuadSPI module is a programmable sequence engine that works on "instruction-operand" pairs. The core controller executes each programmed instruction sequentially. The complete list of instructions and the corresponding operands is given in the following table.

Table 30-427. Instruction set

Instruction	Instruction encoding	Pins	Operand	Action on Serial Flash(es)
CMD	6'd1	N=2'd{0,1,2,3} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads 2'd3- Eight pads	8 bit command value	Provide the serial flash with operand on the number of pads specified
ADDR	6'd2		Number of address bits to be sent (e.g 8'd24 => 24 address bits required)	Provide the serial flash with address cycles according to the operand on the number of pads specified. The actual address to be provided will be derived from the incoming address in case of AHB initiated transactions and the value of SFAR in case of IPS initiated transactions.
DUMMY	6'd3		Number of dummy clock cycles (should be <= 64 cycles)	Provide the serial flash with dummy cycles as per the operand. The PAD information defines the number of pads in input mode. (e.g one pad implies that pad 1 is not driven, rest all are driven)
MODE	6'd4		8 bit mode value	Provide the serial flash with 8 bit operand on the number of pads specified
MODE2	6'd5	N=2'd{0,1}	2 bit mode value	Provide the serial flash with 2 bit operand on the number of pads[1] specified

Table continues on the next page...

Table 30-427. Instruction set (continued)

Instruction	Instruction encoding	Pins	Operand	Action on Serial Flash(es)
MODE4	6'd6	N=2'd{0,1,2}	4 bit mode value	Provide the serial flash with 4 bit operand on the number of pads[2] specified
READ	6'd7	N=2'd{0,1,2,3} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads 2'd3- Eight pads	Read data size in bytes. (For AHB transactions the user's application should ensure that data size is a multiple of 8 bytes)	Read data from flash on the number of pads specified. The data size may be overwritten by writing to the ADATSZ field of the QSPI_BUFxCR registers for AHB initiated transactions and IDATSZ field of IP Configuration Register (QuadSPI_IPCR) for IP initiated transactions.
WRITE	6'd8		Write data size in bytes	Write data on number of pads sepcified. The data size may be overwritten by writing to the IDATSZ field of IP Configuration Register (QuadSPI_IPCR) register
JMP_ON_CS	6'd9	NA	Instruction number	Every time the CS is deasserted, jump to the instruction pointed to by the operand. This instruction allows the programmer to specify the behavior of the controller when a new read transaction is initiated following a CS deassertion.

Table continues on the next page...

Table 30-427. Instruction set (continued)

Instruction	Instruction encoding	Pins	Operand	Action on Serial Flash(es)
ADDR_DDR	6'd10	N=2'd{0,1,2,3} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads 2'd3- Eight pads	Number of address bits to be sent (e.g 8'd24 => 24 address bits required)	Provide the serial flash with address cycles according to the operand on the number of pads sepcified at each clock edge of serial flash clock. The actual address to be provided will be derived from the incoming address in case of AHB initiated transactions and the value of QSPI_SFAR in case of IPS initiated transactions.
MODE_DDR	6'd11		8 bit mode value	Provide the serial flash with 8 bit operand on the number of pads specified at each clock edge of serial flash.
MODE2_DDR	6'd12	N=2'd{0}	2 bit mode value	Provide the serial flash with 2 bit operand on the number of pads specified at each clock edge of serial flash[3]
MODE4_DDR	6'd13	N=2'd{0,1}	4 bit mode value	Provide the serial flash with 4 bit operand on the number of pads specified at each clock edge of serial flash[4].

Table continues on the next page...

Table 30-427. Instruction set (continued)

Instruction	Instruction encoding	Pins	Operand	Action on Serial Flash(es)
READ_DDR	6'd14	N=2'd{0,1, 2,3} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads 2'd3- Eight pads	Read data size in bytes. (For AHB transactions the user's application should ensure that data size is in multiple of 8 bytes)	Read data from flash on the number of pads specified at each clock edge of serial flash. The data size may be overwritten by writing to the ADATSZ field of the QSPI_BUFxCR registers for AHB initiated transactions and IDATSZ field of IP Configuration Register (QuadSPI_IPCR) for IP initiated transactions
WRITE_DDR	6'd15		Write data size in bytes	Write data on the number of pads specified at each clock edge of serial flash. The data size may be overwritten by writing to the IDATSZ field of IP Configuration Register (QuadSPI_IPCR) register
DATA_LEARN	6'd16		8 bit Data learning pattern	Find the correct sampling point with the data learning pattern. When this instruction is encountered, the QSPI_SMPR[DDRSMP] values are ignored and the controller finds the correct sampling point on its own by sampling the data learning pattern.[5]
STOP	8'd0	NA	NA	Stop execution; deassert CS

1. For a one pad instruction, MODE2 will take 2 serial flash clock cycles on the flash interface.
2. For a one pad instruction, MODE4 will take 4 serial flash clock cycles on the flash interface. For a 4 pad instruction, MODE4 will take 1 serial flash clock cycle on the flash interface.
3. For a one pad instruction, MODE2_DDR will take 1 serial flash clock cycle on the flash interface.
4. For a one pad instruction, MODE4_DDR will take 2 serial flash clock cycles on the flash interface. For a 4 pad instruction MODE4_DDR will take half a cycle on the serial flash interface.
5. It is not recommended to have 0x00 or 0xFF as the data learning pattern.

A sequence of such instruction-operand pairs may be pre-populated in the LUT according to the device connected on board. Each instruction-operand pair is of 16 bits (2 bytes) each. Every sequence pre-programmed in the LUT is referred to by its index.

The programmable sequence engine allows the user to configure the QuadSPI module according to the serial flash connected on board. The flexible structure is easily adaptable to new command/protocol changes from different vendors.

30.5.3.2 Flexible AHB buffers

In order to reduce the latency of the reads for AHB masters, the data read from the serial flash is buffered in flexible AHB buffers. There are four such flexible buffers. The size of each of these buffers is configurable with the minimum size being 0 Bytes and maximum size being the size of the complete buffer instantiated.The size of buffer 0 is defined as being from 0 to QSPI_BUF0IND. The Size of buffer 1 is from QSPI_BUF0IND to QSPI_BUF1IND, buffer2 is from QSPI_BUF1IND to QSPI_BUF2IND and buffer 3 is from QSPI_BUF2IND to the size of the complete buffer, which is given in the Chip Configuration Chapter.

Each flexible AHB buffer is associated with the following

1. An AHB master. Optionally, buffer3 may be configured as an "all master" buffer by setting the QSPI_BUF3CR[ALLMST] bit. When buffer3 is configured in such a way, any access from a master not associated with any other buffer is routed to buffer3.
2. A datasize field representing the amount of data to be fetched from the flash on every "missed" access.

The master port number of every incoming request is checked and the data is returned/fetched into the corresponding associated buffer. Every "missed" access to the buffer causes the controller to clear the buffer and fetch QSPI_BUFxCR[ADATSZ] amount of data from the serial flash.As such, there is no benefit in configuring a buffer size of greater than ADATSZ, as the locations greater than ADATSZ will never be used. For any AHB access, the sequence pointed to by the QSPI_BFGENCR[SEQID] field is used for the flash transaction initiated. The data is returned to the master as soon as the requested amount is read from the serial flash. The controller however, continues to prefetch the rest of the data in anticipation of a next consecutive request. Figure 30-414 shows the flexible AHB buffers.

The QSPI_BFGENCR[SEQID] field points to an index of the LUT. Refer to Look-up Table for details.

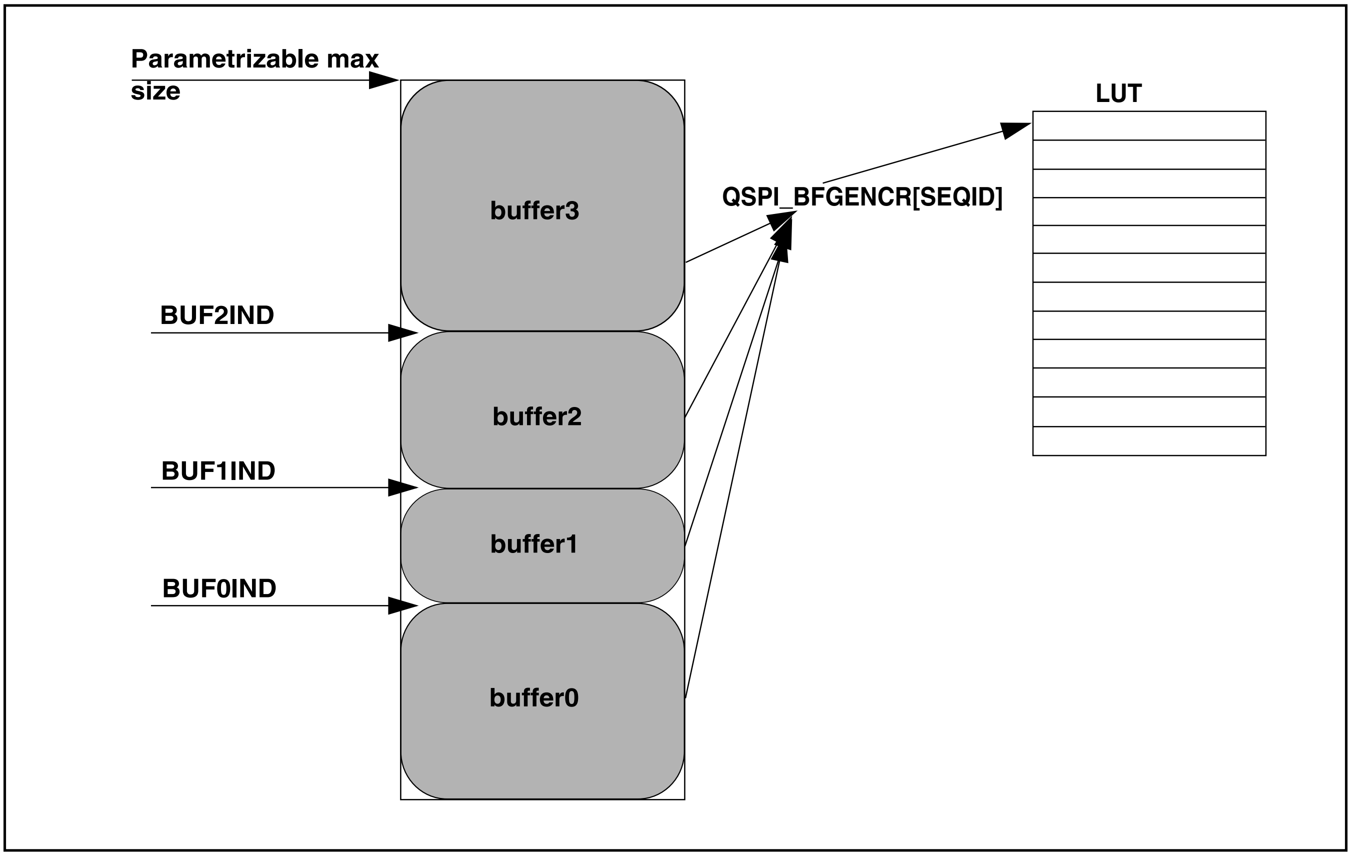


Figure 30-414. Flexible AHB Buffers

Buffer0 may optionally be configured to be associated with a high priority master by setting the QSPU_BUF0CR[HP_EN] bit. An access by a high priority master suspends any ongoing prefetch to any of the other buffers. The ongoing prefetch is suspended and the high priority master is serviced first. Once the high priority masters access completes, the suspended transaction is resumed (before any other AHB access is entertained). The status of the suspended buffer can be read from Sequence Suspend Status Register (QuadSPI_SPNDST).

30.5.3.3 Suspend-Abort Mechanism

Any low priority AHB access can be suspended by a high priority AHB master request. The ongoing transaction is suspended at 64 bit boundary. The suspended transaction is restarted after the high priority master is served and the high priority transaction including data prefetch is completed. While a transaction is in suspended state, it may be aborted if a transaction by the same suspended master is made to a location which is different from the location of the suspended transaction.

Any ongoing transaction is aborted if a request from the same master arrives for a location other than the location at which the transaction is going on. The abort can happen at any point of time.

30.5.3.4 HBURST Support

QuadSPI controller supports HBURST and HSIZE on the AHB interface. HBURST indicates if the transfer forms part of a burst. Four, eight and sixteen beat bursts are supported and the burst may be either incrementing or wrapping. HSIZE indicates the size of the transfer. 8, 16, 32 and 64 bit data size are supported. In case of WRAP accesses, QuadSPI generates aligned accesses to Serial Flash if there is no buffer hit for any incoming non-sequential AHB access. In case there is a buffer hit, the incoming address in the haddr line is latched as it is. If the total burst size is more than the data prefetch size an error response is generated and QSPI_FR[AIBSEF] is set. The data perfetch size can be defined by QSPI_BUFxCR[ADATSZ] or data size mentioned in the sequence pointed to by the SEQID field when ADATSZ is programmed as zero. A few examples are shown in the figure below:

HADDR = 0x38
HBUST = WRAP4
HSIZE = 64 bits
Flash xsaction start = 0x20
Incoming AHB access= 0x38, 0x20, 0x28, 0x30

HADDR = 0x38
HBUST = INCR4
HSIZE = 64 bits
Flash xsaction start = 0x38
Incoming AHB access= 0x38, 0x40, 0x48, 0x50

HADDR = 0x50
HBUST = WRAP8
HSIZE = 64 bits
Flash xsaction start = 0x40
Incoming AHB access= 0x50, 0x58, 0x60, 0x68, 0x70, 0x78, 0x40, 0x48

HADDR = 0xD0
HBUST = WRAP16
HSIZE = 64 bits
Flash xsaction start = 0x80
Incoming AHB access= 0xD0, 0xD8, 0xE0, ...0xF8, 0x80, 0x88, ... 0xC8

HADDR = 0xD4
HBUST = WRAP8
HSIZE = 32bits
Flash xsaction start = 0xC0
Incoming AHB access= 0xD4, 0xD8, 0xDC, 0xC0, 0xC4, 0xC8,0xCC, 0xD0

HADDR = 0x54
HBUST = INCR8
HSIZE = 32bits
Flash xsaction start = 0x54
Incoming AHB access= 0x54, 0x58, 0x5C, 0x60, 0x64, 0x68,0x6C, 0x70

Figure 30-415. QuadSPI HBURST support

NOTE

The software must take care that the prefetch size should never be set less than the minimum data needed by any external interface to start processing.

30.5.3.5 Look-up Table

The Look-up-table or LUT consists of a number of pre-programmed sequences. Each sequence is basically a sequence of instruction-operand pairs which when executed sequentially generates a valid serial flash transaction. Each sequence can have a maximum of 8 instruction-operand pairs. The LUT can hold a maximum of 16 sequences. The figure below shows the basic structure of the sequence in the LUT.

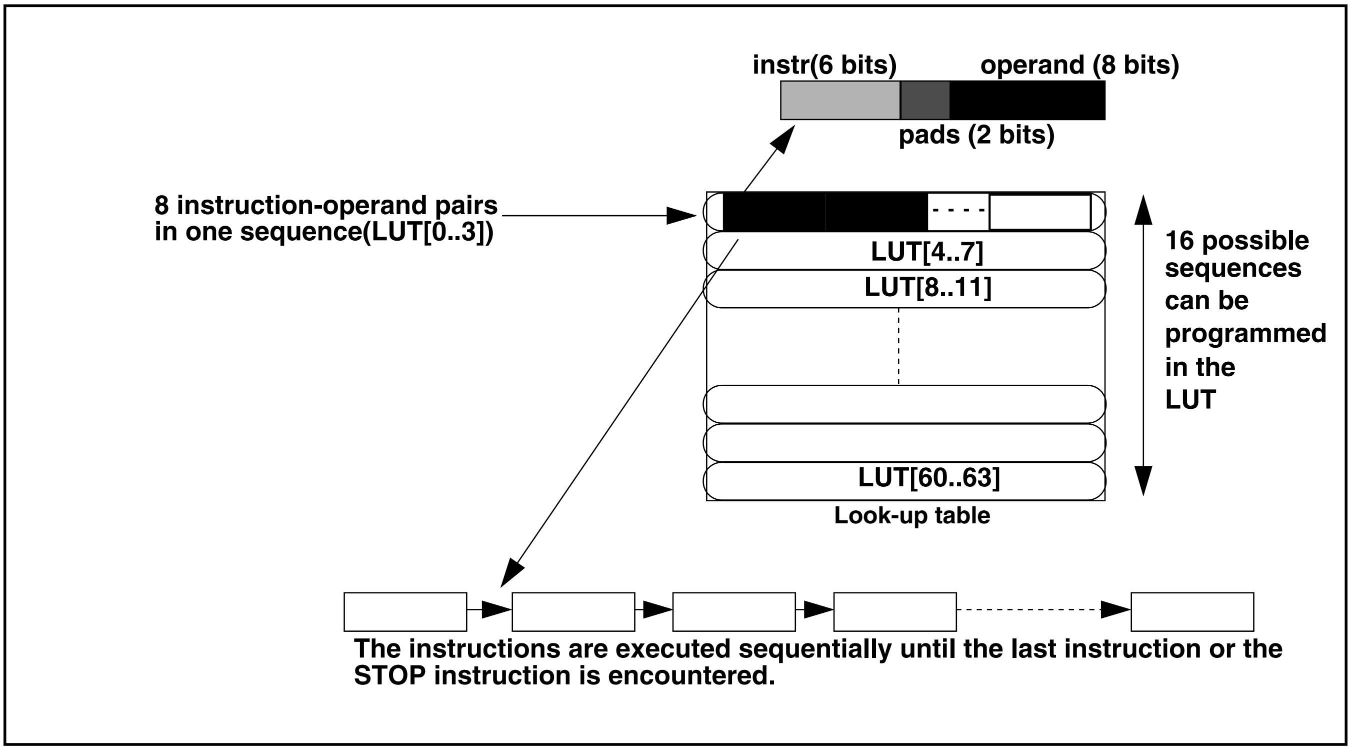


Figure 30-416. LUT and sequence structure

At reset, the index 0 of the look-up-table (LUT[0..3]) is programmed with a basic read sequence as given in Table 30-428. After reset the complete LUT may be reprogrammed according to the device connected on board.In order to protect its contents during a code runover the LUT may be locked, after which a write to the LUT will not be successful until it has been unlocked again.The key for locking or unlocking the LUT is **0x5AF05AF0**. The process for locking and un-locking the LUT is as follows:

Locking the LUT

1. Write the key **(0x5AF05AF0)** in to the LUT Key Register (QuadSPI_LUTKEY).
2. Write 0b01 to the LUT Lock Configuration Register (QuadSPI_LCKCR). Note that this IPS transaction should immediately follow the above IPS transaction (no other IPS transaction can be issued in between). A successful write into this register locks the LUT.

Unlocking the LUT

1. Write the key **(0x5AF05AF0)** into the LUT Key Register (QuadSPI_LUTKEY)
2. Write 0b10 to the LUT Lock Configuration Register (QuadSPI_LCKCR). Note that this IPS transaction should immediately follow the above IPS transaction (no other IPS transaction can be issued in between). A successful write into this register locks the LUT.

The lock status of the LUT can be read from QSPI_LCKCR[UNLOCK] and QSPI_LCKCR[LOCK] bit.

Some example sequences are defined in Example Sequences. The reset sequence at LUT index 0 is given in the following table.

Table 30-428. Reset sequence

Instruction	Pad	Operand	Comment
CMD	0x00	0x03	Read Data byte command on one pad
ADDR	0x00	0x18	24 Addr bits to be sent on one pad
READ	0x00	0x08	Read 64 bits
JMP_ON_CS	0x00	0x00	Jump to instruction 0 (CMD)

30.5.3.6 Issuing SFM Commands

Each access to the external device follows the same sequence:

1. The user must pre-populate the LUT with the serial flash command sequences that are required for the flash device being used.
2. The QuadSPI module starts executing the instructions in the sequence one by one. The transaction starts and the status bit QSPI_SR[BUSY] is set.
3. Communication with the external serial flash device is started and the transaction is executed.

4. When the transaction is finished (all transmit- and receive operations with the external serial flash device are finished) the status bit QSPI_SR[BUSY] is reset. In case of an IP Command the QSPI_FR[TFF] flag is asserted.

Further details are given in below in Flash Programming and Flash Read.

You can trigger the processing of SFM commands in the QuadSPI module in one of the following ways:

- **Using IP commands**

  For IP Commands the required components need to be written into the following registers:

  - Write the serial flash address to be used by the instruction into QSPI_SFAR, refer to Serial Flash Address Register (QSPI_SFAR). For IP Commands not related to specific addresses, the base address of the related flash need to be programmed.For example, for an instruction which does not require an address (i.e. write enable instruction) the SFAR should be programmed with the base address of the memory the command is to be sent to.
  - Write the sequence ID and data size details in the IP Configuration Register (QSPI_IPCR).
  - Note that the write into the QSPI_IPCR[SEQID] field must be the last step of the sequence. It is possible to combine all fields of the QSPI_IPCR into one single write. Refer to IP Configuration Register (QSPI_IPCR) for details.

  Note that there are some conditions where no IP Command is executed after writing the QSPI_IPCR[SEQID] field and the write operation itself is ignored. They are described in Command Arbitration.

- **Using AHB commands**

  Any AHB memory mapped access is routed to one of the buffers depending on the master port number of the request. If the access is a "miss", a new serial flash transaction is started. The transaction is based on the sequence pointed to by the BFGENCR[SEQID] field as described in Flexible AHB buffers.

  An AHB access is termed memory mapped when the access is to the memory mapped serial flashes, as described in Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A and Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B.

  Again the possible error conditions are described in Command Arbitration .

30.5.3.7 Flash Programming

In all cases the memory sector to be written needs to be erased first. The programming sequence itself is then initiated in the following way:

1. Check that the TX Buffer is empty. If the QSPI_SR[TXNE] bit is set then the TX Buffer must be cleared by writing 1 into the QSPI_MCR[CLR_TXF] bit.
2. Program the address related to the command in the QSPI_SFAR register.
3. Provide initial data for the program command into the circular buffer via register TX Buffer Data Register (QSPI_TBDR) . At least one word of data must be written into the TX Buffer up to a maximum of 16.
4. Program the QSPI_IPCR register to trigger the command. The QSPI_IPCR[SEQID] should point to an index of the LUT which has the flash program sequence pre-programmed.The IDATSZ field should be set to denote the size of the write.
5. Depending on the amount of data required, step 3 must be repeated until all the required data have been written into the QSPI_TBDR register.The QSPI_SR[TXFULL] can be used to check if the buffer is ready to receive more data. At any time, the QSPI_TBSR[TRCTR] field can be read to check how many words have been written actually into the TX Buffer.

Upon writing the QSPI_IPCR[SEQID] field (refer to step 4) the QuadSPI module will start to execute the programmed sequence. It is the responsibility of the software to ensure that a correct sequence is programmed into the LUT in accordance with the flash memory connected to the module. The data is fetched from the TX Buffer. It consists of **16** entries of 32-bits and is organized as a circular FIFO, whose read pointer is incremented after each fetch. When all data are transmitted, the QuadSPI module will return from 'busy' to 'idle'. However, this is not true for the external device since the internal programming is still ongoing. It is up to the user to monitor the relevant status information available from the serial flash device and to ensure that the programming is finished properly.

30.5.3.8 Flash Read

Host access to the data stored in the external serial flash device is done in two steps: First the data must be read into the internal buffers and in the second step these internal buffers can be read by the host.

1. **Reading Serial Flash Data into the QuadSPI Module Internal Buffers**

A read access to the external serial flash device can be triggered in two different ways:

- **IP Command Read**: For **reading flash data into the RX Buffer** the user must provide the correct sequence ID in the QSPI_IPCR[SEQID] register. The sequence ID points to a sequence in the LUT. It is the responsibility of the software to ensure that a correct read sequence in programmed in the LUT in accordance with the serial flash device connected on board. The user should program the Serial Flash Address Register (QSPI_SFAR) and the IP Configuration Register (QSPI_IPCR) registers. All available read commands supported by the external serial flash are possible.

  Optionally it is possible to clear the RX Buffer pointer prior to triggering the IP Command by writing a 1 into the QSPI_MCR[CLR_RXF] bit.

  From these inputs, the complete transaction is built when the QSPI_IPCR[SEQID] field is written. The transaction related to the read access starts and the requested number of bytes is fetched from the external serial flash device into the RX Buffer. Since the read access is triggered by an IP command the IP_ACC status bit and the BUSY bit are both set(both are located in the Status Register (QSPI_SR)). A count of the number of entries currently in the Rx Buffer can be obtained from QSPI_RBCT[RXBRD].

  The communication with the external serial flash is stopped when the specified number of bytes has been read (successful completion of the transaction).

- **AHB Command Read**: For **reading flash data into the AHB Buffer** the user must setup a read access by a master to the address range in the system memory map which the external serial flash devices are mapped to. The user should also program the buffer registers corresponding to the AHB master initiating the request, this is depends on the configuration of the QSPI_RBCT[RXBRD]. The user should provide the correct sequence ID into the buffer generic configuration register (QSPI_BFGENCR). It is the responsibility of the software to ensure that a correct read sequence in programmed in the LUT in accordance with the serial flash device connected on board. Flash device selection and access mode are determined by the address accessed in the AHB address space associated to the QuadSPI module, refer to Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A, Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B and Parallel Flash Mode.

  On each AHB read access to the memory mapped area the valid data in the AHB Buffer is checked against the address requested in the actual read. When the AHB read request can't be served from the content of the AHB Buffer, the buffer is flushed and the sequence pointed to by the sequence ID is executed by the

controller. The requested number of buffer entries defined in the QSPI_BUFxCR[ADATSZ] field is then fetched from the external serial flash device into the internal AHB Buffer. Since the read access is triggered via the AHB bus, the QSPI_SR[AHB_ACC] status bit is set driving in turn the QSPI_SR[BUSY] bit until the transaction is finished. The communication with the external serial flash is stopped when the specified number of entries has been filled.

2. **Data Transfer from the QuadSPI Module Internal Buffers**

   The data read out from the external serial flash device by the QuadSPI module is stored in the internal buffers.The means of accessing the data from the buffer differs depending on which buffer the data has been loaded to. Refer to Figure 30-1 for details about the two available buffers, the RX Buffer and the AHB Buffer, in the QuadSPI module:

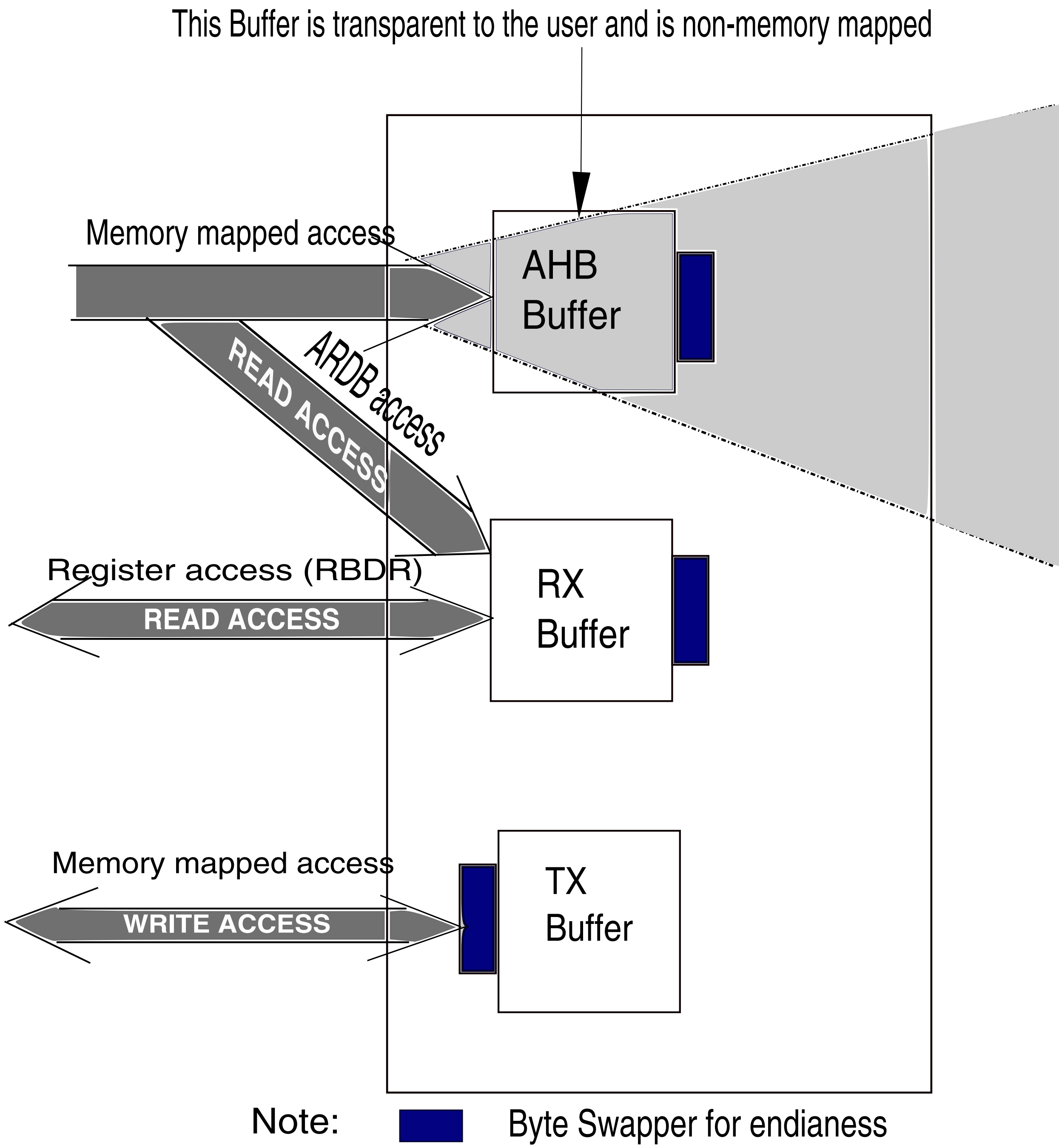


Figure 30-417. QuadSPI memory map

- The RX Buffer is implemented as FIFO of depth 32 entries of 4 bytes. Its content is accessible in two different address areas both referring to the identical data and the same physical memory.

  In the IPS address space in the area associated to QSPI_RBDR0 to QSPI_RBDR31

  In the AHB address space in the area associated to QSPI_ARDB0 to QSPI_ARDB31. Two successive entries are accessed with one single 64 bit AHB read operation.

RX Buffer operation can be summarized as follows: The QSPI_RBCT[WMRK] field determines at which fill level the RXWE bit is asserted and how many entries are removed from the RX Buffer on each Buffer POP operation. So the QSPI_SR[RXWE] bit indicates that the configured number of data entries is available in the RX Buffer and the QSPI_RBSR[RDBFL] field indicates how many valid entries are available in total. Note that the first entry (QSPI_RBDR0 or QSPI_ARDB0) always corresponds to the first valid entry in the RX Buffer. The software needs to manage the number of valid data bytes itself.

Further details can be found in RX Buffer Data Registers 0 - 31 (QSPI_RBDR0 - QSPI_RBDR31) and inAHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31).

- **Flag-based Data Read of the RX Buffer** is done by polling the QSPI_SR[RXWE] bit. When it is asserted the valid entries can be read either via the IPS address space (QSPI_RBDRn) or the AHB address space (QSPI_ARDBn). A Buffer POP operation must be triggered by the application by writing a 1 into the QSPI_FR[RBDF] bit - this automatically updates the FIFO to point to the next entry as defined by RBCT[WMRK]. E.g. if WMRK is set to 3, then the buffer will discard 16bytes of data.
- **DMA controlled Data Read of the RX Buffer** is done by using the DMA module.The application must ensure that the DMA controller of the related device is programmed appropriately like it is described in DMA Usage.

  DMA controlled read out is triggered fully automatically by the assertion of the QSPI_SR[RXWE] bit. The related Buffer POP operation is also handled completely inside the QuadSPI module. Like in the case above, accessing the RX Buffer content either on QSPI_RBDRn or QSPI_ARDBn related addresses is equivalent.

- **AHB Buffer data read via memory mapped access:** This kind of access is done by reading one of the addresses assigned to the external serial flash device(s) within the range given in Table 30-387 table *under the condition that the data requested are already present in the AHB Buffer or it is currently being read from the serial flash device by the instruction in progress.* If this is not the case a memory mapped AHB command read is triggered as described above. If the requested data is already available in the AHB Buffer they are provided directly to the host.

  When AHB access are made to the flash memory mapped address, the data will be fetched and returned to the AHB interface. Till the data is being fetched the AHB interface would be stalled. As soon as the data from the requested address has been read by the QuadSPI module the AHB read access is served. So it is

possible to run sequential reads from the AHB buffer at arbitrary speed without the need to monitor any information about the availability of the data. Nevertheless this access scheme stalls the AHB bus for the time required to read the data from the serial flash device. A better way is (when it is known the access is sequential) to have a prefetch enabled (by programming the ADATSZ field) such that before the next sequential AHB access come, the data is already fetched into the buffer.

As long as the host restricts its accesses to the data already in the buffer and the data currently fetched from the serial flash, it is possible to run the host read from the AHB Buffer in parallel to the serial flash read into the AHB Buffer.

30.5.3.9 Byte Ordering of Serial Flash Read Data

In this paragraph the byte ordering of the serial flash data is given. The basic scheme is that the **first** byte read out of the serial flash device - which is addressed by the QSPI_SFAR[SFADR] field - corresponds to bit position QSPI_RBDR0[31:24] register for IP Command read. In contrast to that for AHB Command read the bytes are always positioned according to the byte ordering of the AHB bus.

- **Byte Ordering in Individual Flash Mode**

  The following table gives the byte ordering scheme of how the byte oriented data space of the serial flash device is mapped into one single 32 bit entry of the RX Buffer or the AHB Buffer. The table is valid within the following context:

  - Flash A or Flash B in Individual Flash Mode
  - All AHB data read commands with access size of 32 bit

Table 30-429. Byte Ordering in Individual Flash Mode

Serial Flash Byte Numbering	3	2	1	0
Buffer Entry Bit Position [31:0] **(32 Bit data width)**	[31:24]	[23:16]	[15:8]	[7:0]

Note

For IP Commands the read size can be given in number of bytes. If this number is not a multiple of 4, then the last buffer entry is not completely filled with the missing higher numbered bytes at undefined values.

For AHB Commands, reads, starting from an address not aligned to 32 bit boundaries, the requested bytes are given at the appropriate positions according to the AMBA AHB specification.

- **Byte Ordering in Parallel Flash Mode**

  In Parallel Flash Mode each byte is combined out of 2 half bytes which are read or written in parallel from the two serial flash devices. The following tables shows how the flash content is separated into the half bytes and how the half bytes are assembled to the content of the QSPI_RBDR0 register.

Table 30-430. Serial Flash Device Half Byte Ordering

Serial Flash Device Byte #	Flash A Bit Position		Flash B Bit Position	
	[7:4]	[3:0]	[7:4]	[3:0]
0	fah0	fal0	fbh0	fbl0
1	fah1	fal1	fbh1	fbl1
2	fah2	fal2	fbh2	fbl2
3	fah3	fal3	fbh3	fbl3
4	fah4	fal4	fbh4	fbl4
5	fah5	fal5	fbh5	fbl5
6	fah6	fal6	fbh6	fbl6
7	fah7	fal7	fbh7	fbl7
8	fah8	fal8	fbh8	fbl8

The table entry naming reflects the half byte positioning in the serial flash devices:

- **<fa>**h0 means **Flash A**, <fb>h0 means Flash B.

- fa**<h>**0 means half byte in **high position**, fa<l>o means half byte in low position.
- fah**<0>** means **physical byte address 0** in the serial flash device, fal<1> means physical byte address 1 in the serial flash device.

Table 30-431. Byte Ordering in Parallel Flash Mode - RX Buffer

QSPI_SFAR[SFADR] set to 0x000_0000								
QSPI_RBDR0 **QSPI_ARDB0**	fal1	fbl1	fah1	fbh1	fal0	fbl0	fah0	fbh0
QSPI_RBDR1 **QSPI_ARDB1**	fal3	fbl3	fah3	fbh3	fal2	fbl2	fah2	fbh2
QSPI_SFAR[SFADR] set to 0x000_0001								
QSPI_RBDR0 **QSPI_ARDB0**	fal2	fbl2	fah2	fbh2	fal1	fbl1	fah1	fbh1
QSPI_RBDR1 **QSPI_ARDB1**	fal4	fbl4	fah4	fbh4	fal3	fbl3	fah3	fbh3

Note

For IP Commands the read size can be given in number of bytes. If this number is not a multiple of 4 the last buffer entry is not completely filled with the missing higher numbered bytes at undefined values.

Table 30-432. Byte Ordering in Parallel Flash Mode - TX Buffer*

QSPI_SFAR[SFADR] set to 0x000_0000								
QSPI_TBDR0	fal1	fbl1	fah1	fbh1	fal0	fbl0	fah0	fbh0
QSPI_TBDR1	fal3	fbl3	fah3	fbh3	fal2	fbl2	fah2	fbh2

* Appicable only for single io mode.

Table 30-433. Byte Ordering in Parallel Flash Mode - AHB Buffer

AHB Address **(32 Bit Access)**	fal1	fbl1	fah1	fbh1	fal0	fbl0	fah0	fbh0
AHB Address 0x800_0004 **(32 Bit Access)**	fal3	fbl3	fah3	fbh3	fal2	fbl2	fah2	fbh2

Note

For AHB Command read starting from an address not aligned to 32 bit boundaries or AHB access size smaller than 32 bit the requested bytes are given at the appropriate positions according to the AMBA AHB specification.

- **Buffer Entry Ordering for 64 Bit Read Access**

  For read access via the AHB interface 64 bit access is possible. Each 64 bit access reads 2 32 bit entries simultaneously. The ordering of these 32 bit entries within the 64 bit word is given in the following table.

Table 30-434. 64 Bit Read Access Buffer Entry Ordering

AHB Read Data Bit Position [63:0]	[63:32]	[31:0]
Buffer Entry #	**Odd (1, 3, 5, ...)**	**Even (0, 2, 4, ...)**

30.5.3.10 Normal Mode Interrupt and DMA Requests

The QuadSPI module has 13 different flags that can only generate interrupt requests and one flag that can generate interrupt as well as DMA requests. The following table lists the eight conditions. Note that the flags mentioned in the table are related to the Flag Register (QSPI_FR).

Table 30-435. Interrupt and DMA Request Conditions

Condition	Flag(QSPI_SPISR)	DMA
Data Learn patttern Failure	DLPFF	-
TX Buffer Fill	TBFF	-
TX Buffer Underrun	TBUF	-

Table continues on the next page...

Table 30-435. Interrupt and DMA Request Conditions (continued)

Condition	Flag(QSPI_SPISR)	DMA
Illegal Instruction Error	ILLINE	-
RX Buffer Drain	RBDF	X
RX Buffer Overflow	RBOF	-
AHB Buffer Overflow	ABOF	-
AHB Sequence Error	ABSEF	-
AHB Illegal Transaction Error	AITEF	-
IP Command Usage Error	IUEF	-
IP Command Trigger during AHB Access Error	IPAEF	-
IP Command Trigger could not be executed Error	IPIEF	-
IP Access during AHB Grant Error	IPGEF	-
IP Command related Transaction Finished	TFF	-

Each condition has a flag bit in the Flag Register (QSPI_FR) and a Request Enable bit in the DMA Request Select and Enable Register (QSPI_RSER) . The RX Buffer Drain Flag (RBDF) has separate enable bits for generating IRQ and DMA requests. Note that not all flags have an individual IRQ line. Check the devices Interrupt Vector Table for more details.

- Transmit Buffer Fill Interrupt Request:

  The Transmit Buffer Fill IRQ indicates that the TX Buffer can accept new data. It is asserted if the QSPI_FR[TBFF] flag is asserted and if the corresponding enable bit (QSIP_RSER[TBFIE]) is set. Refer to TX Buffer Operation, for details about the assertion of the QSPI_FR[TBFF] flag.

- Receive Buffer Drain Interrupt or DMA Request:

  The Receive Buffer Drain IRQ derived from the QSPI_FR[RBDF] flag indicates that the RX Buffer of the QuadSPI module has data available from the serial flash device to be read by the host. It remains set as long as the QSPI_RBSR[RXWE] bit is set. The QSPI_RSER[RBDIE] bit enables the related IRQ.

  Aside from the IRQ it is possible to handle RX Buffer drain by DMA. If the QSPI_RSER[RBDDE] bit is set, a DMA request will be triggered when the RX Buffer contains more than QSPI_RBCT[WMRK] valid entries.The application must set the environment appropriately (e.g. the DMA controller) for the DMA transfers.

- Buffer Overflow/Underrun Interrupt Request:

The Buffer Overflow/Underrun IRQ is a combination of the following flags (all located in the QSPI_FR register with the related enable bits in the QSPI_RSER register):

- TBUF - TX Buffer Underrun, enabled by TBUIE
- RBOF - RX Buffer Overflow, enabled by RBOIE
- ABOF - AHB Buffer Overflow, enabled by ABOIE

  The Transmit Buffer Underrun indicates that an underrun condition in the TX Buffer has occurred. It is generated when a write instruction is triggered whilst the Tx Buffer is empty and the QSPI_RSER[TFUFIE] bit is set.

  The Receive Buffer Overflow indicates that an overflow condition in the RX Buffer has occurred. It is generated when the RX Buffer is full, an additional read transfer attempts to write into the RX Buffer and the QSPI_RSER[RBOIE] bit is set.

  The AHB Buffer Overflow indicates that an overflow condition in the AHB Buffer has occurred. It is generated when the AHB Buffer is full, an additional read transfer attempts to write into the AHB Buffer and the QSPI_RSER[ABOIE] bit is set.

  The data from the transfers that generated the individual overflow conditions is ignored.

- Serial Flash Command Error Interrupt Request

  If the IPAEF, IPIEF, IPGEF, ABSEF or IUEF flags in the QSPI_SR are set, and the related interrupt enable bits in the QSPI_RSER are also set, then an interrupt is requested.

- Transaction Finished Interrupt Request

  The IP Command Transaction Finished IRQ indicates the completion of the current IP Command.It is triggered by the QSPI_FR[TFF] flag and is masked by the QSPI_RSER[TFIE] bit.

30.5.3.11 TX Buffer Operation

The TX Buffer provides the data used for page programming. For proper operation it is required to provide at least one entry in the TX Buffer prior to starting the execution of the page programming command. The application must ensure that the required number of data bytes is written into the TX Buffer fast enough as long as the command is executed without a TX Buffer overflow or underrun.

The QuadSPI module sets the QSPI_FR[TBFF] flag so long as the TX Buffer is not full and can accept more data.

When the QuadSPI module tries to pull data out of an empty TX Buffer the TX Buffer underrun is signaled by the QSPI_FR[TBUF] flag. The current IP Command leading to the underrun condition is continued until the specified number of bytes has been sent to the serial flash device, in the underrun condition, the data transferred is undefined i.e. once the underrun flag is set, it will return the garbage value until the required number of bytes are not sent. When this Sequence Command is finished, the QSPI_FR[TBFF] flag is asserted indicating that the Tx Buffer is ready to be written again.

The TX Buffer overflow isn't signaled explicitly, but the TX Buffer fill level can be monitored by the QSPI_TBSR[TRBFL] field.

Refer to TX Buffer Status Register (QuadSPI_TBSR) and Flag Register (QuadSPI_FR) for details about the TX Buffer related registers.

30.5.3.12 Address scheme

Earlier serial flash memories supported only 24-bit address space hence restricting the maximum memory size of the serial flash as 16 MB. The new memory specification supports two types of 32-bit addressing mode in addition to legacy 24-bit address mode.

- **Extended Address Mode**

  In this mode, the legacy 24-bit commands are converted to accept 32-bit address commands. The flash memory needs to be configured for 32-bit address mode. Also, while programming the LUT sequence in QuadSPI for 32-bit mode, the ADDR and ADDR_DDR command should be programmed with 8’d32 as the operand value. By default, the QuadSPI is in 24-bit legacy address mode. Each of the memory vendors have a different way of enabling this mode (Refer to the memory specification from memory vendors). For example, the command B7h sent to Macronix flash will enable it for 32-bit address mode.

- **Extended Address register**

  In this mode, the upper 8-bit of the 32-bit address is provided by the Extended address register in the memory itself. The memory provides a specific register which is updated according to the address to be accessed. This effectively converts the legacy 24-bit address command into 32-bit address commands. The memories greater in size than 16 MB, consists of banks of 16 MB. The 8-bit written in the extended address register effectively enables a bank. For example in Spansion memory, when the extended address register is updated with a value of 0x01 with the help of the command 17h, it will open Bank1 of the memory. The consequent 24-bit address

commands will lead to Bank1. The extended address register needs to be update with the respective value for access to other banks. This effectively converts the legacy 24-bit address command into 32-bit address commands.

30.6 Initialization/Application Information

This section provides the initialization and application information of the QuadSPI module.

30.6.1 Power Up and Reset

Note that the serial flash devices connected to the QuadSPI module may require special voltage characteristics of their inputs during power up or reset. It is the responsibility of the application to ensure this.

30.6.2 Available Status/Flag Information

This paragraph gives an overview of the different status and flag information available and their interdependencies for different use cases. Related registers are QSPI_SR and QSPI_FR. Refer to the related descriptions how to set up the QuadSPI module appropriately.

30.6.2.1 IP Commands

Refer to IP Configuration Register (QuadSPI_IPCR) for additional details not explicitly covered in this paragraph.

- **IP Commands - Normal Operation**

  Writing the QSPI_IPCR[SEQID] field triggers the execution of a new IP Command. Given that this is a legal command the QSPI_SR[IPACC] and the QSPI_SR[BUSY] bits are asserted simultaneously, immediately after the execution is started.

  When the instruction on the serial flash device has been finished these bits are de-asserted and the QSPI_FR[TFF] flag is set.

- **IP Commands - Error Situations**

  Refer to Table 30-436 below.

30.6.2.2 AHB Commands

Refer to Section 1, Reading Serial Flash Data into the QuadSPI Module, in Flash Read for additional details not explicitly covered in this paragraph.

- **AHB Commands - Normal Operation**

  Memory mapped read access to a serial flash address not contained in the AHB Buffer, triggers the execution of an AHB Command. Given that this is a legal command the QSPI_SR[AHBACC] and the QSPI_SR[BUSY] bits are asserted simultaneously immediately after the execution is started. When the instruction on the serial flash device has been finished these bits are de-asserted.

- **IP Commands - Error Situations**

  Refer to Table 30-436 below.

30.6.2.3 Overview of Error Flags

The following table gives an overview of the different error flags in the QSPI_FR register and additional error-related details.

Table 30-436. Overview of QSPI_FR Error Flags

Error Category	Error Flag in QSPI_FR	Command Execution on Serial Flash Device TFF Behavior (in case of IP commands only)	Description
AHB Error Flags	ABSEF	Flash transaction is aborted	AHB sequence contains • WRITE instruction • WRITE_DDR instruction
	ABOF	Flash transaction continues until it finishes	Set when the module tried to push data into the AHB buffer that exceeded the size of the AHB buffer. Only occurs due to wrong programming of the QSPI_BUFxCR[ADATSZ].
Miscellaneous Error Flags	DLPFF	Flash transaction continues until it finishes	Set when DATA_LEARN instruction was encountered in a sequence but no sampling point was found for the data learning pattern.
	ILLINE	Flash transaction aborted	Illegal instruction Error Flag – Set when an illegal instruction is encountered by the controller in any of the sequences.

Table continues on the next page...

Table 30-436. Overview of QSPI_FR Error Flags (continued)

Error Category	Error Flag in QSPI_FR	Command Execution on Serial Flash Device TFF Behavior (in case of IP commands only)	Description
Command Arbitration Errors	IPIEF	TFF not asserted in conjunction with that command	IP Command Error - caused when IP access is currently in progress (IP_ACC set) and • write attempt to QSPI_IPCR register. • write attempt to QSPI_SFAR register. • write attempt to QSPI_RBCT register.
	IPAEF		• AHB Command already running, another IP Command could not be executed. • AHB Command already running, write attempt to QSPI_IPCR[SEQID] field.
	IPGEF		• Exclusive access to the serial flash granted for AHB Commands, write attempt to QSPI_IPCR[SEQID] field.
IP Command Error	IUEF		• IP Command Usage Error
Buffer Related Errors	RBOF	TFF is asserted on completion	• RX Buffer Overrun
	TBUF		• TX Buffer Underrun

Note that only the buffer related errors are related to a transaction on the external serial flash. All the other errors do not trigger an actual transaction.

30.6.2.4 IP Bus and AHB Access Command Collisions

There are two flags related to this topic, the QSPI_FR[IPAEF] and QSPI_FR[IPIEF]. Refer to sub-section "Reading Serial Flash Data into the QuadSPI Module" of Flash Read section, for a description of the flags and Command Arbitration , for details about possible command collisions.

30.6.3 Exclusive Access to Serial Flash for AHB Commands

It is possible that several masters need to access the serial flash device connected to the QuadSPI module separately, one master by triggering IP Commands and reading the RX Buffer (via RBDx register) and the other masters by triggering AHB Commands (via ARDBx Registers). These two set of buffer (RBDR and ARDB Buffer) points to the same physical buffer.Refer to Figure 30-417 To avoid command collisions resulting in excessive latencies the QuadSPI module implements a request-handshake mechanism between the master triggering AHB Commands and the QuadSPI module allowing this specific master to request exclusive access to the serial flash device for AHB Commands. If this exclusive access is granted the execution of IP Commands is blocked. This resolves command collisions and excessive times where the AHB interface may be blocked.

If this capability is used in the device there is additional status and flag information available related to this mechanism. The QSPI_SR[AHBGNT] bit reflects the module-internal state that the exclusive access mentioned above is granted, any attempt to trigger an IP Command is rejected and results in the assertion of the QSPI_FR[IPGEF] flag. Refer to the descriptions of the related bit and flag for details.

It is within the responsibility of the application to set up the master using this mechanism appropriately, if used incorrectly no IP Commands at all can be triggered.

Two different cases can be distinguished:

30.6.3.1 RX Buffer Read via QSPI_ARDB Registers

In this case all masters share the AHB bus for RX Buffer as well as for AHB Buffer read. In this case the access to the AHB interface by the master triggering AHB Commands must be deferred until any pending IP Command has been finished **and** the RX Buffer readout has been finished as well. The QSPI ARDB Buffers access the Rx buffer i.e the data from the Rx Buffer is returned and no data from AHB Buffer is touched. This is the conservative use case, corresponding to the reset value 0 of the QSPI_RBCT[RXBRD] bit.

In this case the QSPI_SR[AHBGNT] bit is asserted not earlier than any running IP Command has been finished (QSPI_SR[IP_ACC] is 0), the RX Buffer has been read out completely (QSPI_RBSR[RDBFL] equal to 0) or no DMA read is pending (QSPI_SR[RXDMA] equal to 0 and Rx Buffer readout is via AHB(QSPI_RBCT[RXBRD]) equal to 1.

30.6.3.2 RX Buffer Read via QSPI_RBDR Registers

This is the preferred use case as an access to the AHB buffer (memory mapped flash) does not interfere with any IPS access to read the RBDR buffer. It is not possible that a pending AHB bus access triggered by an AHB Command stalls the AHB bus and blocks the RX Buffer readout since the RX Buffer is read via the IP bus based registers QSPI_RBDR0 to QSPI_RBDR31.

For this case it is recommended to program the QSPI_RBCT[RXBRD] bit to 1. The QSPI_SR[AHBGNT] bit is asserted immediately after any running IP Command has been finished (QSPI_SR[IP_ACC] is 0), the RX Buffer has been read out completely (QSPI_RBSR[RDBFL] equal to 0) or no DMA read is pending (QSPI_SR[RXDMA] equal to 0, allowing the master triggering AHB Commands to trigger AHB Commands as soon as possible without the need to wait for the RX Buffer readout to be finished.

30.6.4 Command Arbitration

In case of overlapping commands, the arbitration scheme is described in the following paragraphs under the assumption that the priority mechanism described in Exclusive Access to Serial Flash for AHB Commands is **not** used:

- During the execution of an IP Command, the running IP Command can't be terminated by issuing another IP Command or AHB Command. The QSPI_FR[IPIEF] flag is asserted when the host tries to write into the QSPI_IPCR register. When the host triggers an AHB Command (refer to sub-section "Reading Serial Flash Data into the QuadSPI Module" of Flash Read section, for details), this command is stalled until the currently running IP Command is finished.
- During the execution of an AHB Command, the running AHB Command can't be terminated by issuing an IP Command. The command is ignored and the QSPI_FR[IPAEF] flag is asserted. Refer to Flag Register (QuadSPI_FR) for the description of these flags.

  When another AHB Command is triggered the address of the memory mapped access is considered. If the requested address is currently read from the serial flash device, the running command is continued. If this is not the case the currently running command is terminated and another AHB Command related to the requested address is executed. Refer to sub-section "Reading Serial Flash Data into the QuadSPI Module" of Flash Read section, for further details.

In case of coinciding commands the IP Command is triggered and the AHB Command is stalled until the IP Command has been finished (QSPI_SR[IP_ACC] has been deasserted).

The IP Commands ignored in case of command collision will not result in the assertion of the QSPI_FR[TFF] flag.

30.6.5 Flash Device Selection

Regardless of the SFM Command (IP or AHB) the access mode is selected by specifying the 32 bit address value for the following SFM Command.

For IP Commands the access mode is selected with the address programmed into the QSPI_SFAR register. Refer to Serial Flash Address Register (QuadSPI_SFAR) for details.

For AHB Commands the access mode is determined by the memory mapped address which is accessed Refer to AMBA Bus Register Memory Map for details.

30.6.6 DMA Usage

For the complete description of the DMA module refer to the related DMA Controller chapter. In this paragraph only the details specific to the DMA usage related to the QuadSPI module are given.

30.6.6.1 DMA Usage in Normal Mode

30.6.6.1.1 Bandwidth considerations

Careful consideration of the throughput rate of the entire chain (serial flash -> AHB bus / IP Bus -> DMA controller) involved in the read data process is essential for proper operation. Such analysis must take into account not only the data rate provided by the serial flash but also the data rate of the AHB bus and the performance of the DMA controller in reading data from the RX buffer.

Two figures must match for proper operation, that means that the data rate provided by the serial flash device must not exceed the average RX Buffer readout data rate. Otherwise, the longer this state persists, a RX Buffer overflow will result.

AHB Bus Side:

The total number of bus cycles for each DMA Minor Loop completion is added from the following components:

- Overhead for each minor loop, given by DMA controller: Assume 10 cycles

- Overhead due to clock domain crossing: Assume 2 cycles
- Number of bus clock cycles required for 8 bytes (64 bit read size): Assume 2 cycles (read/write sequence of DMA controller)

Note that the size of the minor loop is determined by the size of the QSPI_RBCT[WMRK] field, therefore the overhead given above distributes among (QSPI_RBCT[WMRK]+1)/2 read accesses of 64 bit each.

The following table gives some examples for typical use cases:

Table 30-437. Access Duration Examples - Bus Clock Side

QSPI_RBCT[WMRK]	Number of Bytes per DMA Loop [1]	Number of Bus Clock cycles for DMA Minor Loop	Time Duration of DMA Minor Loop for 120Mhz Bus clock Frequency
0	4	12+2 = 14	~117ns
1	8	12+2 = 14	~117ns
3	16	12+4 = 16	~133ns
7	32	12+8 = 20	~167ns
11	48	12+12 = 24	~200ns

1. DMA Loop means one Minor Loop Completion which is equivalent to one.

Serial Flash Device Side:

The number of serial flash cycles can be determined in the following way:

- Number of serial flash clock cycles required to read 4 bytes, corresponding to one RX Buffer entry(setup of command and address not considered): 2 cycles for Quad DDR mode instructions in Parallel Flash Mode, 4 cycles for Quad(SDR) mode instruction in parallel flash mode or Dual IO DDR mode instruction in parallel flash mode, 8 cycles for Quad Mode (SDR) instructions in Individual Flash Mode etc.
- Overhead due to clock domain crossing : 1 cycle.

The following table lists the number of clock cycles required to read the data from the serial flash corresponding to the different settings of the QSPI_RBCT[WMRK] field:

Table 30-438. Access Duration Examples - Serial Flash side

QSPI_RBCT[WMRK] setting	Num Bytes per DMA Loop [1]	Num SCKFx for 60MHz SCKFx			Time duration of Flash data readout for 60MHz SCKFx (~16.6ns period)		
		IFM [2] Quad	IFM Quad DDR	[3] PFM Quad DDR	IFM Quad	IFM Quad DDR	PFM Quad DDR
0	4	9	5	3	~150ns	~83ns	~50ns
1	8	17	9	5	~282ns	~150ns	~83ns
3	16	33	17	9	~548ns	~282ns	~150ns
7	32	65	33	17	~1079ns	~548ns	~282ns
11	48	97	49	25	~1610ns	~813ns	~415ns

1. DMA Loop means one Minor loop completion which is equivalent to one Major Loop iteration.
2. Individual flash mode.
3. Parallel flash mode.

From the examples given in the two tables above, it can be seen that depending on the relationship between the Bus clock and Serial flash clock frequencies, there are settings possible where the serial flash provides the read data faster than the AHB bus can read out the RX buffer. In the above tables, it is the case of PFM Quad DDR mode with Watermark up to 3 and other cases. In these cases, the RX buffer data keeps accumulating over time and will eventually overflow. To avoid RX Buffer overflow, the data transaction size should be small enough.

A complementary example would be when the watermark is set to be too high. In such a case, the time taken by the DMA to read out the RX buffer entries should be smaller than the time taken by the controller to push in the remaining entries in the buffer.

30.6.7 Parallel mode

QuadSPI can access two flashes in parallel. This increases the throughput of the QuadSPI by two times. Only read operations and x1 mode write are allowed in parallel mode. In case a write transaction is initiated in with mode than one pad in parallel mode, QSPI_FR[IUEF] is set. When dual die flashes are accessed in parallel mode, it is mandatory for flash A1 to be of the same size as B1 and A2 to be of the same size as B2. The following figure shows how QuadSPI maps the incoming addresses to the different flashes connected on board.

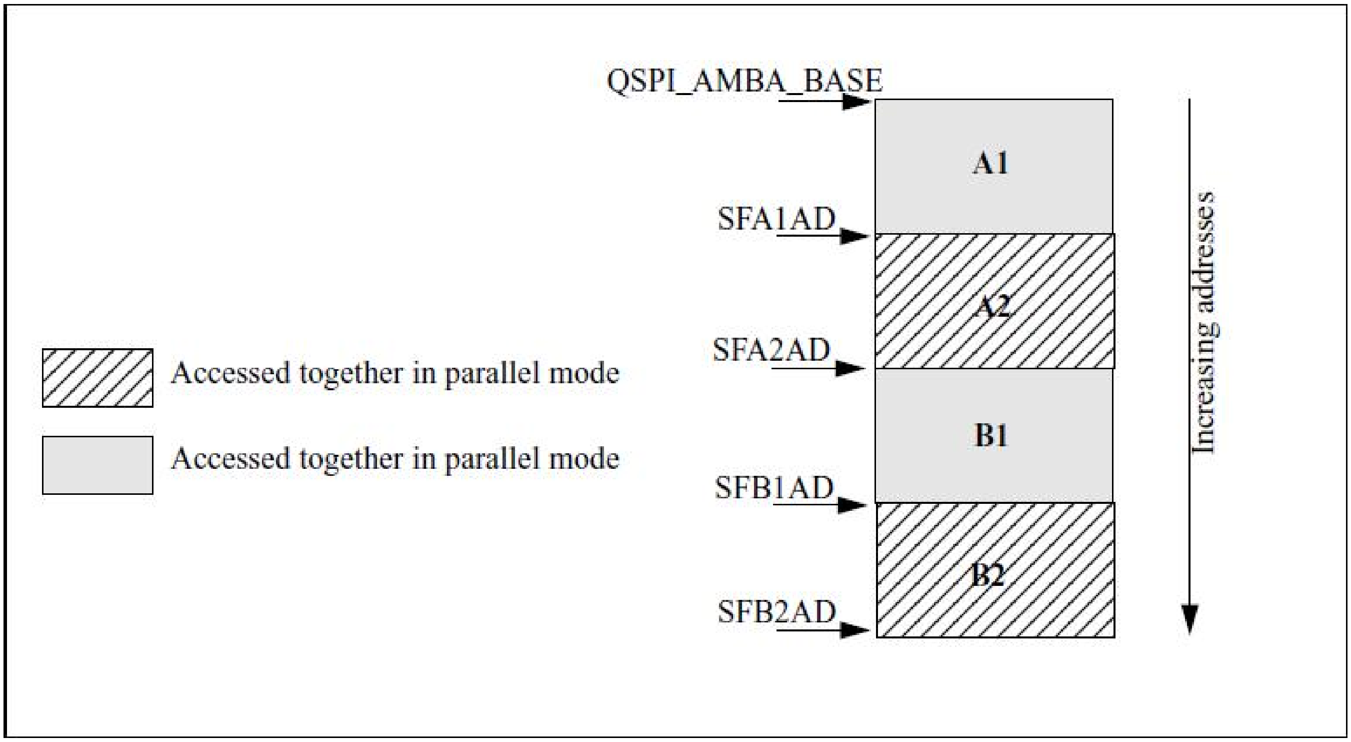


Figure 30-418. Flash addressing

An example programming for parallel mode access is given below (flash sizes are assumed to be 256MB):

- QSPI_AMBA_BASE - 0x10000000
- QSPI_SFA1AD[TPADA1] - 0x20000000
- QSPI_SFA2AD[TPADA2] - 0x30000000
- QSPI_SFB1AD[TPADB1] - 0x40000000
- QSPI_SFB2AD[TPADB2] - 0x50000000

In order to access the first location of A1/B1 pair, the incoming address should be 0x10000000. QSPI_AMBA_BASE is subtracted from this address and the result is divided by two. Therefore, address provided to flash A1 and B1

Flash Address = (Memory mapped address - QSPI_AMBA_BASE)/2

For Memory Mapped address:

- 0x10000000, flash address: 0x0 (Or, the first address of flash A1 and B1)
- 0x10000004, flash address: 0x2
- 0x10000008, flash address: 0x4 etc.

Similarly, in order to access the first location of A2/B2 pair, the incoming address should be 0x30000000.

Flash Address = (Memory mapped address - SFA2AD)/2

For Memory Mapped address:

- 0x30000000, flash address: 0x0 (Or, the first address of flash A2 and B2)
- 0x30000004, flash address: 0x2
- 0x30000008, flash address: 0x4 etc.

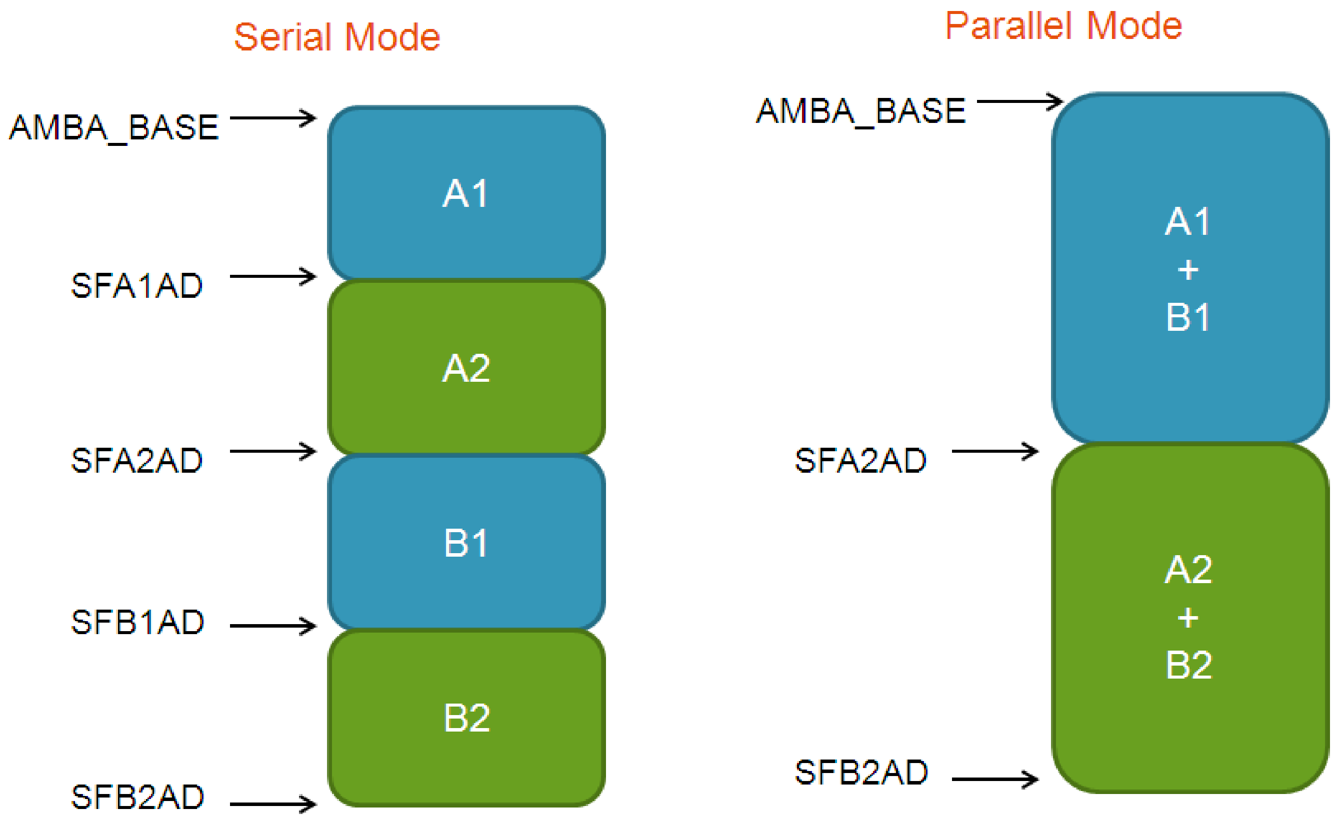


Figure 30-419. Memory map - Serial and Parallel

30.7 Byte Ordering - Endianess

QuadSPI provides support for swapping the flash read/write data based on the configuration of the QSPI_MCR[END_CFG]. By default the data is always returned in 64bit BE format on the AHB bus and 32bit BE format on the IPS interface when read via the RX buffer and written in 32bit BE format when written via the TX buffer.

The table(QSPI_MCR[END_CFG]) below shows the complete bit ordering. BE signifies Big Endian which means the high order bits of the associated data vectors are associated with low order address positions. LE signifies Little Endian which means the lower order bits of the associated data vectors are associated with low order address positions.Refer to figure Figure 30-417

Table 30-439. QSPI_MCR[END_CFG]

00	64 bit BE
01	32 bit LE
10	32 bit BE
11	64 bit LE

The tables below (Byte ordering configuration in AHB) and (Byte ordering configuration in IPS) show how this configuration is implemented in QSPI AHB and IPS interfaces respectively. B in the table signifies Byte and the index 1-8 refers to the byte position i.e. 1 refer to bits[7:0], 8 refer to bits[63:56] and so on.

Table 30-440. Byte ordering configuration in AHB

64 bit BE	B1	B2	B3	B4	B5	B6	B7	B8
64 bit LE	B8	B7	B6	B5	B4	B3	B2	B1
32 bit BE	B5	B6	B7	B8	B1	B2	B3	B4
32 bit LE	B4	B3	B2	B1	B8	B7	B6	B5

Table 30-441. Byte ordering configuration in IPS

32BE	B1	B2	B3	B4
32LE	B4	B3	B2	B1

The examples below show the byte ordering in 64 bit BE configuration for AHB Buffer and 32 bit BE for TX/RX Buffer:

30.7.1 Programming Flash Data

CPU write instructions to the QSPI_TBDR register like

- Write QSPI_TBDR -> 0x01_02_03_04
- Write QSPI_TBDR -> 0x05_06_07_08

result in the following content of the TX Buffer:

Table 30-442. Example of QuadSPI TX Buffer

TX Buffer Entry	Content
0	32'h01_02_03_04
1	32'h05_06_07_08

Programming the TX Buffer into the external serial flash device results in the following byte order to be sent to the serial flash:

- 01…02…03…04…05…06…07…08

30.7.2 Reading Flash Data into the RX Buffer

Reading the content from the same address provides the following sequence of bytes, identical to the write case:

- 01…02…03…04…05…06…07…08

This results in the RX Buffer filled with:

Table 30-443. Resulting RX Buffer Content

RX Buffer Entry	Content
0	32'h01_02_03_04
1	32'h05_06_07_08

30.7.2.1 Readout of the RX Buffer via QSPI_RBDRn

The RX Buffer content appears at CPU read access via the IP SkyBlue interface in the following order:

- Read QSPI_RBDR0 <- 0x01_02_03_04
- Read QSPI_RBDR1 <- 0x05_06_07_08

30.7.2.2 Readout of the RX Buffer via ARDBn

The RX Buffer content appears at read access on the AMBA AHB interface at the QuadSPI module boundary:

- (1a): 32 Bit Access: Read QSPI_ARDB0 <- 0x01_02_03_04
- (2a): 32 Bit Access: Read QSPI_ARDB1 <- 0x05_06_07_08
- (1b/2b): 64 Bit Access: Read QSPI_ARDB0 <- 0x01_02_03_04_05_06_07_08

30.7.3 Reading Flash Data into the AHB Buffer

Reading the content from the same address as it was written to provides the following sequence of bytes, identical to the write case:

- 01…02…03…04…05…06…07…08

This results in the AHB Buffer filled with:

Table 30-444. Resulting AHB Buffer Content

AHB Buffer Entry	Content
0	64'h01_02_03_04_05_06_07_08

30.7.3.1 Readout of the AHB Buffer via Memory Mapped Read

The AHB Buffer content appears at read access on the AMBA AHB interface at the QuadSPI module boundary:

- (1a): 32 Bit Read Access: <- 0x01_02_03_04
- (2a): 32 Bit Read Access: <- 0x05_06_07_08
- (1/2): 64 Bit Read Access: <- 0x01_02_03_04_05_06_07_08

30.8 Serial Flash Devices

Several different vendors make flash devices with a QuadSPI interface. At present there is no set standard for the QuadSPI instruction set. Most common commands currently have the same instruction code for all vendors, however some commands are unique to specific vendors. Some example sequences are provided below.

30.8.1 Example Sequences

This section provides the example sequences of the QuadSPI module.

Table 30-445. Exit 4 x I/O Read Enhance Performance Mode (XIP) (Macronix) and Read Status

INSTR	PAD	OPERAND	COMMENT
CMD	0x0	0xEB	4xIO Read Command
ADDR	0x2	0x18	24 Bit address to be send on 4 pads
MODE	0x2	0x00	2 mode cycles (exit XIP)
DUMMY	0x0	0x04	4 dummy cycles
READ	0x2	0x08	Read 64 bits
CMD	0x0	0x05	Read Status register
READ	0x0	0x01	Status register data
STOP	0x0	0x00	STOP, Instruction over

30.8.1.1 Fast Read Sequence (Macronix/Numonyx/Spansion/Winbond)

The following table shows the fast read sequence for Macronix/Numonyx/Spansion/Winbond flashes.

Table 30-446. Fast Read sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0x0B	Fast Read command = 0x0B
ADDR	0x0	0x18	24 Addr bits to be sent on one pad
DUMMY	0x0	0x08	8 Dummy cycles
READ	0x0	0x04	Read 32 Bits on one pad
JMP_ON_CS	0x0	0x00	Jump to instruction 0 (CMD)

30.8.1.2 Fast Dual I/O DT Read Sequence (Macronix)

The following table shows the Fast Dual I/O DT read sequence for Macronix flashes.

Table 30-447. Fast Dual I/O DT Read sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0xBD	Fast Dual I/O DT read command = 0xBD

Table continues on the next page...

Table 30-447. Fast Dual I/O DT Read sequence (continued)

Instruction	Pad	Operand	Comment
ADDR_DDR	0x1	0x18	24 Addr bits to be sent on 2 pads in DDR mode
MODE4_DDR	0x1	0x00	P2=P0 or P3=P1 is necessary. Refer to Macronix datasheet for details. One clock cycle for mode.
DUMMY	0x0	0x06	6 Dummy cycles
READ_DDR	0x1	0x04	Read 32 Bits on 2 pads in DDR mode
JMP_ON_CS	0x0	0x00	Jump to instruction 0 (CMD)

30.8.1.3 Fast Read Quad Output (Winbond)

The following table shows the Fast read quad output sequence for Winbond memories

Table 30-448. Fast Read Quad output sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0x6B	Fast read quad output command = 0x6B
ADDR	0x0	0x18	24 Addr bits to be sent on 1 pad
DUMMY	0x0	0x08	8 Dummy cycles
READ	0x2	0x04	Read 32 Bits on 4 pads
JMP_ON_CS	0x0	0x00	Jump to instruction 0 (CMD)

30.8.1.4 4 x I/O Read Enhance Performance Mode (XIP) (Macronix)

The following table shows the 4 x I/O Read Enhance Performance Mode for Macronix flashes. The enhanced performance mode is also known as XIP mode.

Table 30-449. Fast Read Quad output sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0xEB	4xI/O Read command = 0xEB
ADDR	0x2	0x18	24 Addr bits to be sent on 4 pads
MODE	0x2	0xA5	2 mode cycles
DUMMY	0x0	0x04	4 Dummy cycles
READ	0x2	0x04	Read 32 Bits on 4 pads
JMP_ON_CS	0x0	0x01	Jump to instruction 1 (ADDR)

When in XIP mode the software should ensure that all the flashes connected to the controller are in XIP mode. As a part of initializing the controller, all the flashes may be enabled with XIP by carrying out dummy reads.

30.8.1.5 Dual Command Page Program (Numonyx)

The following table shows the Dual command page program sequence for Numonyx flashes.

Table 30-450. Dual Command Page Program sequence

Instruction	Pad	Operand	Comment
CMD	0x1	0x02	Dual command page program = 0x02 on 2 pads
ADDR	0x1	0x18	24 Addr bits to be sent on 2 pads
WRITE	0x1	0x20	Write 32 Bytes on 2 pads
STOP	0x0	0x00	STOP, Instruction over

30.8.1.6 Sector Erase (Macronix/Spansion/Numonyx)

The following table shows the Sector erase sequence for Macronix/Spansion/Numonyx flashes

Table 30-451. Sector Erase sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0xD8	Sector erase command = 0xD8
ADDR	0x0	0x18	24 Addr bits to be sent on 1 pad
STOP	0x0	0x00	STOP, Instruction over

30.8.1.7 Read Status Register (Macronix/Spansion/Numonyx/Winbond)

The following table shows the Read status register sequence for Macronix/Spansion/Numonyx/Winbond flashes.

Table 30-452. Read Status Register Sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0x05	Read status register command = 0x05
READ	0x0	0x01	Read status register data
STOP	0x0	0x00	STOP, Instruction over

30.8.2 Dual Die Flashes

Certain serial flash vendors provide dual-die packages which are essentially two devices (dies) stacked within the same package to increase the memory capacity of a single package. These two devices within a package share the same data and clock pins, but have individual Chip Selects. QuadSPI controller provides support for two dual-die packages to be connected simultaneously. The figure below shows the two dual-die packages and the naming conventions used in this document. For simplicity, the data pins are shown to be unidirectional.

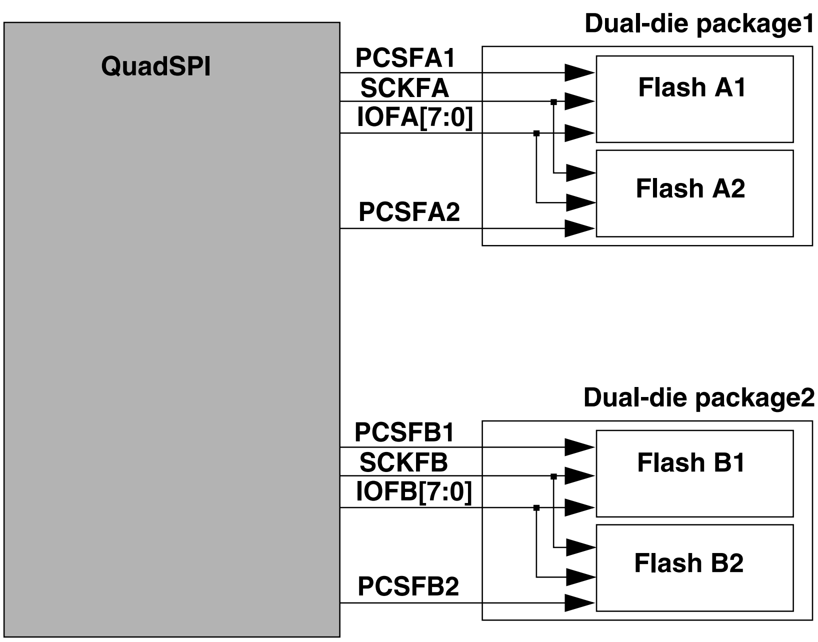


Figure 30-420. Dual-die support

Since the two devices within one package share the same i/o pads, they cannot function in parallel mode. Software should ensure that when QuadSPI is configured in parallel mode the two selected flash devices are from different dual-die packages.

30.8.3 Boot initialization sequence

The following are the recommended sequence of steps for booting from QuadSPI (please refer to #converted-map for details).

- System out of reset and flash available (300us)

- Clocks still at very low frequency. Clock tree configured, I/O pins configured. First request sent to QuadSPI for address 0x0 of flash.
- The reset command sequence in QuadSPI has 0x03 (basic read command) which is applicable to all flashes at < 50MHz serial flash clock
- The first few bytes of data is read from the flash which contains the following information:
  - The total sizes of all the flashes connected on board
  - Whether DDR mode supported
  - Frequency of DDR operation
  - Continuous mode entry sequence
  - 24bit or 32bit addressing (assuming 24bit for first accesses)
- All the serial flashes are configured
  - Quad Mode enabled
  - Dummy reads to enter into XIP
- QuadSPI is configured
  - Parallel enable set
  - LUT configured for highest performance reads
  - DDR mode enabled (if applicable)
  - Buffers configured
- Serial flash clock frequency increased.
- Boot reads happen in parallel, DDR enabled, quad output mode @66MHz.

30.8.4 Serial Flash Clock Frequency Limitations

Certain commands of some serial flash devices are limited in the frequency applied to the serial flash device on command execution. In order to support these commands without having to relibrate the module clocks, the the serial flash device clock can be divided by 2 (half speed) by setting the QSPI_SMPR[HSENA] bit. The SCLK will return to full speed once the the QSPI_SMPR[HSENA] bit is cleared.

30.9 Internal Sampling of Serial Flash Input Data

30.9.1 Internal Sampling of Serial Flash Input Data

Depending from the actual implementation there is a delay between the internal clocking in the QuadSPI module and the external serial flash device.Refer to the following figure for an overview of this scheme.

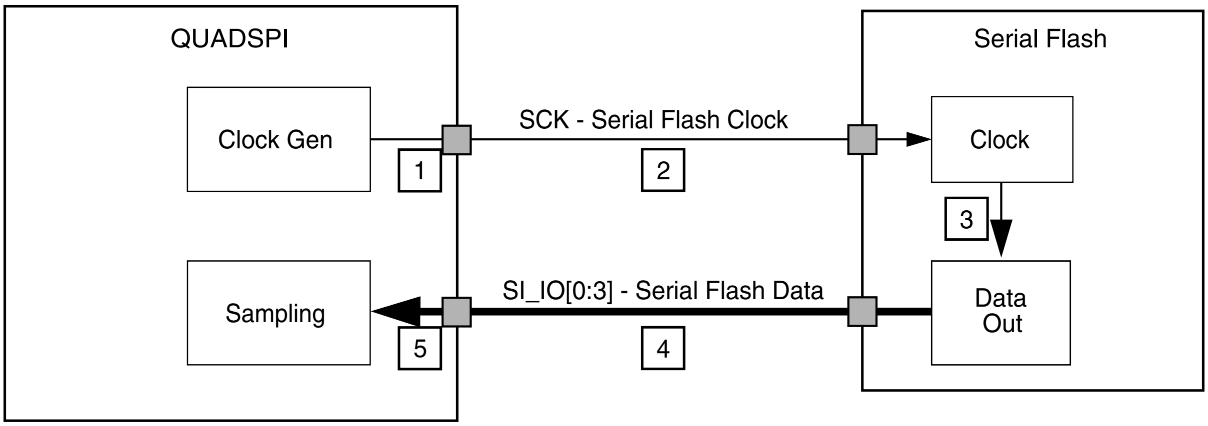


Figure 30-421. Serial Flash Sampling Clock Overview

Note

The arrival of the serial flash data in the sampling stage of the QuadSPI module are given in the following figure. Note that the amount of the total delay $t_{Del,total}$ is very specific to the characteristics of the actual implementation.

Note also that the serial flash device clock SCK is inverted with respect to the QuadSPI internal reference clock.

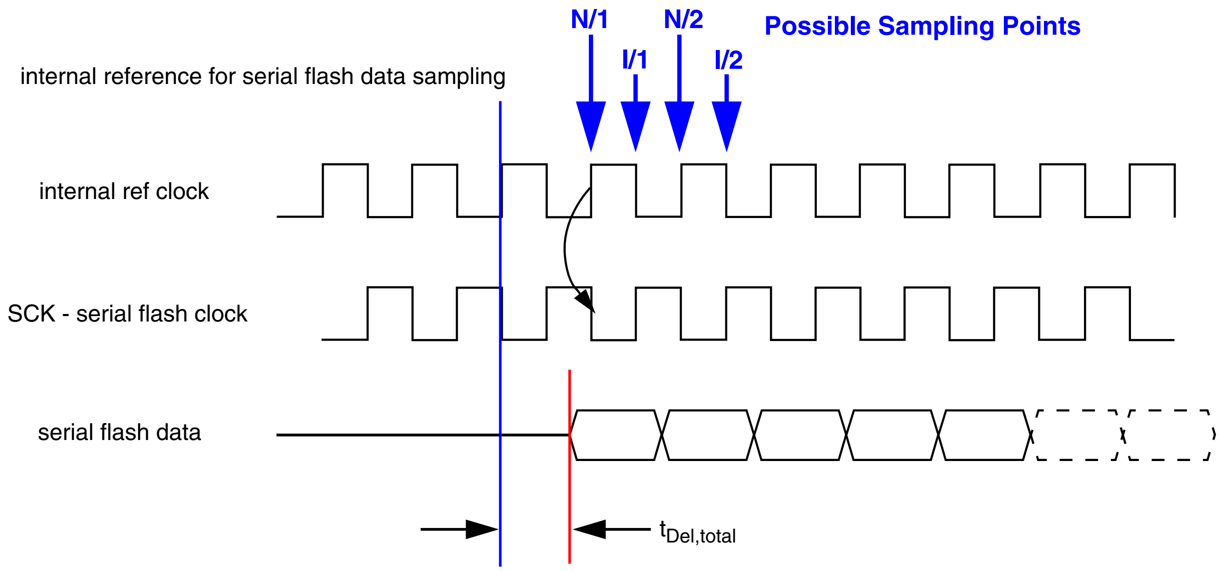


Figure 30-422. Serial Flash Sampling Clock Timing

The rising edge of the internal reference clock is taken as timing reference for the data output of the serial flash. After a time of $t_{Del,total}$ the data arrive at the internal sampling stage of the QuadSPI module.

According to the Serial Flash Sampling Clock Overview figure, the following parts of the delay chain contribute to $t_{Del,total}$:

1. Output delay of the serial flash clock output of the device containing the QuadSPI module
2. Wire delay of application/PCB from the device containing the QuadSPI module to the external serial flash device
3. Clock to data out delay of the external serial flash device, including input and output delays
4. Wire delay of application/PCB from the external serial flash device to the device containing the QuadSPI module
5. Input delay belonging to the data in input

The possible points in time for the sampling of the incoming data are denoted as N/1, I/1, N/2 and I/2 above. The sampling point relevant for the internal sampling is configured in the QSPI_SMPR register, refer to Sampling Register (QuadSPI_SMPR) for details. Note that the falling edges of the reference clock are not actually used, instead the inverted clock is used for sampling at these positions. The following table gives an overview of the available configurations for the commands running at regular (full) speed:

Table 30-453. Sampling Configuration

Sampling Point	Description	Delay [FSDLY] [HSDLY]	Phase [FSPHS] [HSPHS]	QSPI_SMPR for Full Speed Setting[1]
N/1	sampling with non-inverted clock, 1 sample delay	0	0	0x0000000x
I/1	sampling with inverted clock, 1 sample delay	0	1	0x0000002x
N/2	sampling with non-inverted clock, 2 samples delay	1	0	0x0000004x
I/2	sampling with inverted clock, 2 samples delay	1	1	0x0000006x

1. 'x' is not considered here

Depending from the actual delay and the serial flash clock frequency the appropriate sampling point can be chosen. The following remarks should be considered when selecting the appropriate setting:

- Theoretically there should be 2 settings possible to capture the correct data since the serial flash output is valid for 1 clock cycle, disregarding rise and fall times and timing uncertainties.
- Depending from the timing uncertainties it may turn out in actual applications that only one possible sample positions remains. This is subject to careful consideration depending from the actual implementation.
- The delay $t_{Del,total}$ is an absolute size to shift the point in time when the serial flash date get valid at the QuadSPI input.
- For decreasing frequency of the serial flash clock the distance between the edges increases. So for large differences in the frequency the required setting may change.
- For commands running at half of the regular serial flash clock (QSPI_SMPR[HSENA] bit set) the sampling point must be figured separately to allow for the compensation of the absolute shift in time with respect to the sample-relative setting in the QSPI_SPMPR register.

30.9.2 DDR Mode

When the serial flashes function in DDR mode, the data is valid for only half a clock cycle. This, along with the fact that the time for which the data is actually valid is smaller than half a clock cycle, requires that we provide closely spaced sampling points. The QuadSPI samples the incoming data at multiple sampling points provided by a 4x serial

flash clock in DDR mode. The figure below shows the different sampling points as configured by QSPI_SMPR[DDRSMP].The FSDLY/FSPHS and HSDLY/HSPHS bits are ignored for DDR instructions.

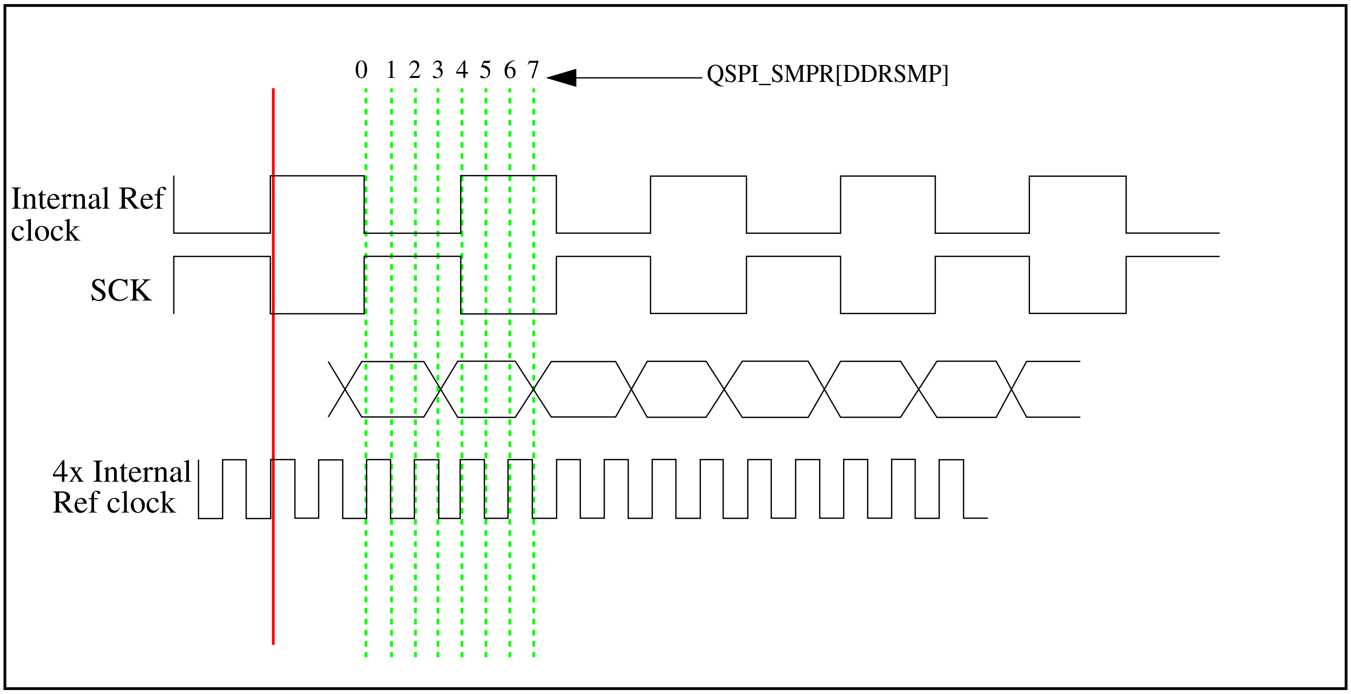


Figure 30-423. DDR sampling edges

Software should ensure that the correct sampling value is configured in the QSPI_SMPR[DDRSMP] register.

30.10 Data Strobe Signal functionality

Some external serial flashes provide the data strobe (DQS/RDS) output which is fed directly to the QuadSPI module. The strobe (DQS/RDS) signal needs to be delayed to have the edges aligned to the data valid period. QuadSPI internally samples the incoming data at posedge of the strobe signal for SDR and on both the edges of the strobe signal for DDR. Refer to the figure for more detail.

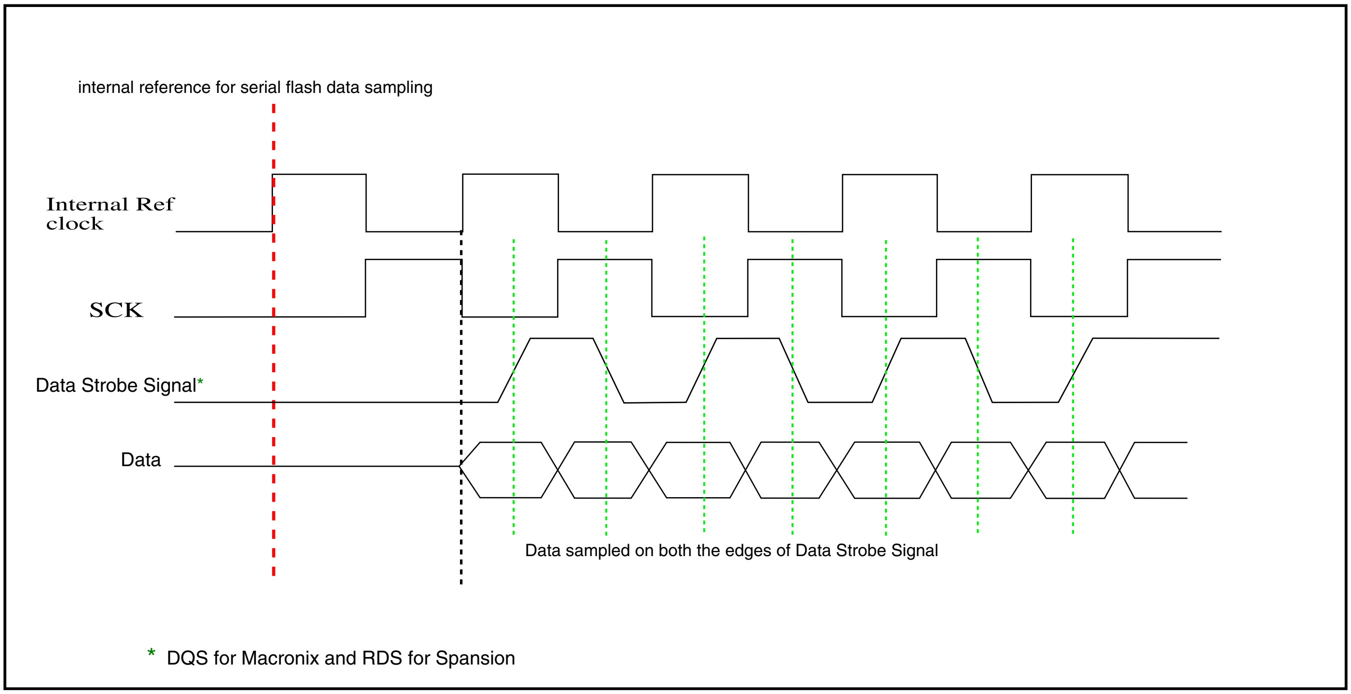


Figure 30-424. Data strobe signal functionality

Chapter 31
NAND Flash Controller (NFC)

31.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The NAND flash controller (NFC) interfaces to standard NAND flash memory devices. It is composed of various control logic units and a 9 KB SRAM buffer. The NFC provides a glueless interface to 8- and 16-bit NAND flash devices with page sizes of 512 bytes, 2 KB, 4 KB, 8 KB and 16 KB.

Throughout this chapter the following terms are used:

- Block — (specified by device) smallest erasable unit in a NAND device, consisting of multiple pages
- Page — (specified by device) unit of flash data containing main and spare areas
- Main area of a page — stores data
- Spare area of a page — stores ECC and other software information
- Sector — an elementary transfer unit
  - For devices with pages of 2KB and smaller, this is the same size of the page
  - For devices with pages larger than 2KB, the pages are split into multiple virtual pages. In this case, the sector size is the size of the virtual page
- Virtual page — is the physical page size divided by the splitting factor, NFC_CFG[PAGECNT]

- ECC — error-correcting code
- BCH (Bose Chaudhuri Hocquenghem) — cyclic error-correcting code that corrects multi-bit errors

31.1.1 Block Diagram

The following is a block diagram of the NAND flash controller.

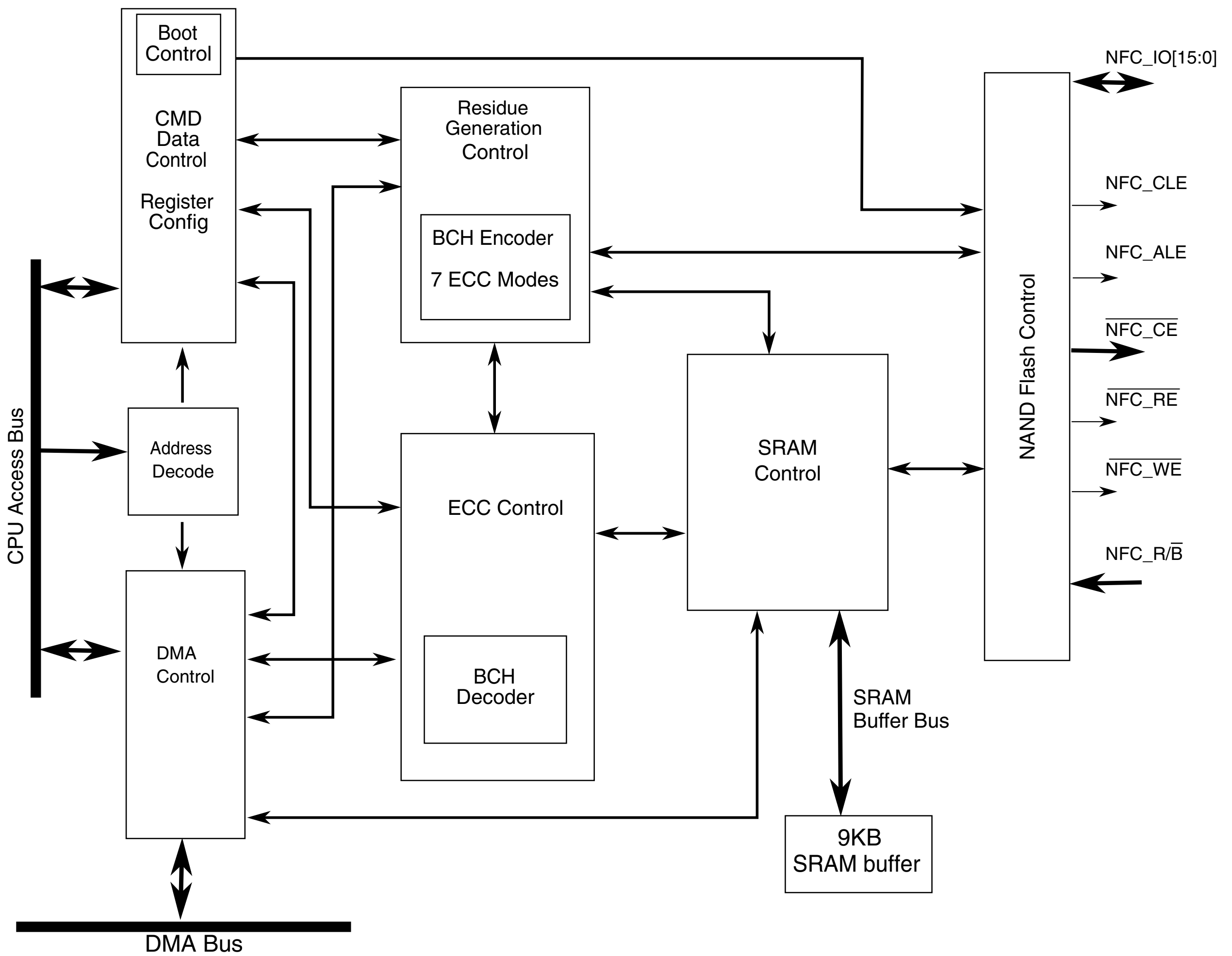


Figure 31-1. NAND Flash Controller Block Diagram

31.1.2 Features

The NAND flash controller includes the following features:

- 8- and 16-bit NAND flash interface

- 9 KB RAM buffer
    - Memory-mapped registers and SRAM buffer
- Supports all NAND flash products regardless of density/organization
- Supports flash device commands, such as page read, page program, reset, block erase, read status, read ID, copy-back, multiplane read/program, interleaved read/program, random input/output, read in EDO mode.
- Integrated DMA engine
    - Two configurable DMA channels
        - Use DMA channel 1 only to read/write a page for main and spare area of a page
        - Use DMA channel 1 to read/write the main area of a page, and DMA channel 2 for the spare area
- ECC mode
    - In ECC mode, NFC supports 4/6/8/12/16/24/32-bit error correction.
    - ECC mode can be bypassed.
- Boot from page size ≥ 2KB flash (x8) without extra control

31.2 External Signal Description

The signals shown in the next table are used to control NAND flash device.

Table 31-1. NFC Signal Properties

Name	Function	I/O	Reset
NFC_ALE	Flash address latch enable	O	1
$\overline{\text{NFC\_CE}}$	Flash chip enable	O	1
NFC_CLE	Flash command latch enable	O	1
NFC_R/$\overline{\text{B}}$	Flash ready/busy	I	Pull up[1]
$\overline{\text{NFC\_RE}}$	Flash read enable	O	1
$\overline{\text{NFC\_WE}}$	Flash write enable	O	1
NFC_IO[15:0]	Flash data bus	I/O	—

1. Need to configure both PE and PS bit to 1'b1 of pin control register PORTC_PCR18 to make NFC_R/B pull up, when PTC18 is configured to MUX=6 (NFC_R/B).

31.3 Memory Map/Register Definition

This section defines the NAND flash controller's registers.

NFC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400E_3F00	Flash command 1 (NFC_CMD1)	32	R/W	30FF_0000h	31.3.1/1300
400E_3F04	Flash command 2 (NFC_CMD2)	32	R/W	007E_E000h	31.3.2/1301
400E_3F08	Column address (NFC_CAR)	32	R/W	0000_0000h	31.3.3/1302
400E_3F0C	Row address (NFC_RAR)	32	R/W	1100_0000h	31.3.4/1302
400E_3F10	Flash command repeat (NFC_RPT)	32	R/W	0000_0000h	31.3.5/1303
400E_3F14	Row address increment (NFC_RAI)	32	R/W	0000_0001h	31.3.6/1304
400E_3F18	Flash status 1 (NFC_SR1)	32	R	0000_0000h	31.3.7/1304
400E_3F1C	Flash status 2 (NFC_SR2)	32	R	0000_0000h	31.3.8/1305
400E_3F20	DMA channel 1 address (NFC_DMA_CH1)	32	R/W	0000_0000h	31.3.9/1305
400E_3F24	DMA configuration (NFC_DMACFG)	32	R/W	0000_0000h	31.3.10/ 1306
400E_3F28	Cach swap (NFC_SWAP)	32	R/W	0FFE_0FFEh	31.3.11/ 1306
400E_3F2C	Sector size (NFC_SECSZ)	32	R/W	0000_0420h	31.3.12/ 1307
400E_3F30	Flash configuration (NFC_CFG)	32	R/W	000E_A631h	31.3.13/ 1308
400E_3F34	DMA channel 2 address (NFC_DMA_CH2)	32	R/W	0000_0000h	31.3.14/ 1310
400E_3F38	Interrupt status (NFC_ISR)	32	R/W	6000_0000h	31.3.15/ 1311

31.3.1 Flash command 1 (NFC_CMD1)

Address: 400E_0000h base + 3F00h offset = 400E_3F00h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BYTE2								BYTE3								0															
W																																
Reset	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_CMD1 field descriptions

Field	Description
31–24 BYTE2	Second command byte that may be sent to the flash device

Table continues on the next page...

NFC_CMD1 field descriptions (continued)

Field	Description
23–16 BYTE3	Third command byte that may be sent to the flash device
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

31.3.2 Flash command 2 (NFC_CMD2)

Address: 400E_0000h base + 3F04h offset = 400E_3F04h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BYTE1								CODE							
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CODE								0					BUFNO		BUSY_START
W																
Reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_CMD2 field descriptions

Field	Description
31–24 BYTE1	First command byte that may be sent to the flash device
23–8 CODE	User-defined flash operation sequencer Each bit indicates a certain action. If the bit is set, the corresponding action is executed after writing 1 to START. The following are some configuration examples (other sequences are possible): 0111_1110_1110_0000 Read data (BYTE1, 5x Address, BYTE2, R/ $\overline{B}$, read data) 1111_1111_1101_1000 Write page (DMA,BYTE1, 5x Address, write data, BYTE2, R/ $\overline{B}$, BYTE3, read status) 0100_1110_1101_1000 Block erase (BYTE1, 3x Address, BYTE2, R/ $\overline{B}$, BYTE3, read status) 0100_1000_0000_0100 Read ID (BYTE1, 1x Address, read ID) 0100_0000_0100_0000 Reset (BYTE1, R/ $\overline{B}$) 0111_1110_0000_0000 CMD+address (BYTE1, 5xaddress) 1111_1111_1100_0000 Write page burst (DMA,BYTE1,5xAddress, write data, BYTE2,R/ $\overline{B}$)
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

NFC_CMD2 field descriptions (continued)

Field	Description
2–1 BUFNO	Internal buffer number used for this command
0 BUSY_START	Busy indicator and start command This busy indicator is repeated in the NFC_ISR register. **NOTE:** Read to this bitfield indicates BUSY whereas write indicates START. 0 During reads, flash controller is idle and it is okay to send next command. During writes, no action. 1 During reads, command execution is busy. During writes, start command execution.

31.3.3 Column address (NFC_CAR)

Address: 400E_0000h base + 3F08h offset = 400E_3F08h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																BYTE2								BYTE1							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_CAR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 BYTE2	Second byte of column address
7–0 BYTE1	First byte of column address

31.3.4 Row address (NFC_RAR)

Address: 400E_0000h base + 3F0Ch offset = 400E_3F0Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			RB0	0		CS1	CS0	BYTE3							
W																
Reset	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BYTE2								BYTE1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_RAR field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 RB0	Ready/busy 0 enable Determines if NFC_R/ $\overline{\text{B}}$ 0 is waited on a wait for R/ $\overline{\text{B}}$ command. If an equal number of $\overline{\text{NFC\_CE}}$ and NFC_R/ $\overline{\text{B}}$ lines are used, the CS *n* and RB *n* fields must contain identical values. If only one NFC_R/ $\overline{\text{B}}$ is used, then CS *n* determines the true chip select, and this field is always 1. 0 NFC_R/ $\overline{\text{B}}$ 0 is disabled 1 NFC_R/ $\overline{\text{B}}$ 0 is enabled
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 CS1	Chip select 1 enable 0 $\overline{\text{NFC\_CE1}}$ is disabled 1 $\overline{\text{NFC\_CE1}}$ is enabled
24 CS0	Chip select 0 enable 0 $\overline{\text{NFC\_CE0}}$ is disabled 1 $\overline{\text{NFC\_CE0}}$ is enabled
23–16 BYTE3	Third byte of row address
15–8 BYTE2	Second byte of row address
7–0 BYTE1	First byte of row address

31.3.5 Flash command repeat (NFC_RPT)

Address: 400E_0000h base + 3F10h offset = 400E_3F10h

Bit	31–16	15–0
R	0	COUNT
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

NFC_RPT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 COUNT	16-bit repeat count Determines how many times NFC_CMD2[CODE] is executed. If 0 or 1, the flash command is executed once.

31.3.6 Row address increment (NFC_RAI)

When auto-increment of row address is enabled (NFC_CFG[AIAD] = 1), the row address is incremented as follows:

```
new{NFC_RAR[BYTE3, BYTE2, BYTE1]} =
                            {NFC_RAR[BYTE3], NFC_RAR[BYTE2], NFC_RAR[BYTE1]} +
{NFC_RAI[INC3, INC2, INC1]}
```

Address: 400E_0000h base + 3F14h offset = 400E_3F14h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								INC3								INC2								INC1							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

NFC_RAI field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 INC3	Increment for the third byte of row address
15–8 INC2	Increment for the second byte of row address
7–0 INC1	Increment for the first byte of row address

31.3.7 Flash status 1 (NFC_SR1)

Address: 400E_0000h base + 3F18h offset = 400E_3F18h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ID1								ID2								ID3								ID4							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_SR1 field descriptions

Field	Description
31–24 ID1	First byte returned by read ID command
23–16 ID2	Second byte returned by read ID command
15–8 ID3	Third byte returned by read ID command
7–0 ID4	Fourth byte returned by read ID command

31.3.8 Flash status 2 (NFC_SR2)

Address: 400E_0000h base + 3F1Ch offset = 400E_3F1Ch

Bit	31–24	23–8	7–0
R	ID5	0	STATUS1
W			
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

NFC_SR2 field descriptions

Field	Description
31–24 ID5	Fifth byte returned by read ID command
23–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 STATUS1	Byte returned by read status command

31.3.9 DMA channel 1 address (NFC_DMA_CH1)

Address: 400E_0000h base + 3F20h offset = 400E_3F20h

Bit	31–0
R W	ADDRESS
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

NFC_DMA_CH1 field descriptions

Field	Description
31–0 ADDRESS	DMA channel 1 address. DMA channel 1 address, it is 8-byte aligned.

31.3.10 DMA configuration (NFC_DMACFG)

Address: 400E_0000h base + 3F24h offset = 400E_3F24h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COUNT1												COUNT2			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT2			OFFSET2				0							ACT1	ACT2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_DMACFG field descriptions

Field	Description
31–20 COUNT1	Number of bytes to be transferred by DMA channel 1. It should be multiple of 8 bytes.
19–13 COUNT2	Number of bytes to be transferred by DMA channel 2. It should be multiple of 8 bytes.
12–9 OFFSET2	256-byte offset for DMA channel 2. DMA channel 2 transfer starts at this offset count x 256 bytes. For example, if OFFSET2 = 0x2, DMA channel 2 transfer starts at 0x200.
8–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 ACT1	DMA channel 1 status 0 Inactive 1 Active, and transfers to memory when triggered
0 ACT2	DMA channel 2 status 0 Inactive 1 Active, and transfers to memory when triggered

31.3.11 Cach swap (NFC_SWAP)

When DMA transfers data to/from the NFC cache (NFC SRAM buffer), or when the CPU reads or writes data to/from the NFC cache via the internal bus, all accesses that go to NFC_SWAP[ADDR1] are directed to NFC_SWAP[ADDR2]. Likewise, all accesses that go to NFC_SWAP[ADDR2] are directed to NFC_SWAP[ADDR1].

The feature allows the bad block marker in the first position of the spare area of a page. Because of the way the flash controller interleaves data and ECC bytes on flash devices with page sizes larger than 2 KB, the position of the bad block marker is shifted, and does not appear in the first position of the spare area of the page. The cache swap feature allows consistent swapping of the actual bad block line with the expected bad block line, and causes the operating system to get the bad block marker in the position where it is expected.

Address: 400E_0000h base + 3F28h offset = 400E_3F28h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				ADDR1											0
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				ADDR2											0
W																
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0

NFC_SWAP field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–17 ADDR1	Lower swap address
16–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–1 ADDR2	Upper swap address
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

31.3.12 Sector size (NFC_SECSZ)

Address: 400E_0000h base + 3F2Ch offset = 400E_3F2Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																			SIZE												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0

NFC_SECSZ field descriptions

Field	Description
31–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

NFC_SECSZ field descriptions (continued)

Field	Description
12–0 SIZE	Size in bytes of one elementary transfer unit For devices with pages of 2KB and smaller, this is the physical size of the page in bytes (data bytes + ECC bytes) transferred in one page. When pages are larger than 2KB, they must be split in multiple virtual pages. In this case, the sector size is the size of the virtual page. The virtual page size is the physical size divided by the splitting factor, NFC_CFG[PAGECNT]. **NOTE:** If only a part of a page to be programmed or read, SIZE can be set to the number of affected bytes, not the page size. Then, ECC and DMA (data bytes) are all performed on the number of bytes, indicated by SIZE. **NOTE:** For 16-bit data width flash devices, only odd SIZE is supported. If SIZE is even number, the real implemented size is SIZE – 1. So, write size + 1 to this field. For example, if SIZE = 1, no data is written or read. **NOTE:** When programming NAND memory for boot and using the ECC feature, ensure that SIZE is equal to number of (data + ECC bytes).

31.3.13 Flash configuration (NFC_CFG)

NOTE

NFC_CFG[BTMD] resets to 0 if no boot is performed by the NFC; 1 if booting from NFC.

Address: 400E_0000h base + 3F30h offset = 400E_3F30h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	STOPWERR	ECCAD[11:3]									ECCSRAM	DMAREQ	ECCMODE			FAST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	IDCNT			TIMEOUT					BITWIDTH	BTMD	AIAD	AIBN	PAGECNT			
Reset	1	0	1	0	0	1	1	0	0	0	1	1	0	0	0	1

NFC_CFG field descriptions

Field	Description
31 STOPWERR	Stopping on write error

Table continues on the next page...

NFC_CFG field descriptions (continued)

Field	Description
	0 No stop on write error 1 Auto-sequencer stops on a write error
30–22 ECCAD[11:3]	Byte address in SRAM where ECC status is written. **NOTE** ECCAD[2:0] are always zeros
21 ECCSRAM	Writing ECC status to SRAM 0 Do not write ECC status to SRAM 1 Write ECC status to SRAM
20 DMAREQ	Transferring sectors after ECC 0 Do not transfer sector after ECC is done 1 After ECC is done, transfer sector using DMA
19–17 ECCMODE	ECC type 000 No correction, ECC bypass 001 4-error correction (8 ECC bytes) 010 6-error correction (12 ECC bytes) 011 8-error correction (15 ECC bytes) 100 12-error correction (23 ECC bytes) 101 16-error correction (30 ECC bytes) 110 24-error correction (45 ECC bytes) 111 32-error correction (60 ECC bytes)
16 FAST	See the "Fast Flash Configuration for EDO" section for more details. 0 Slow flash timing. Clock in read data on rising edge of read strobe 1 Fast flash timing. Clock in read data a half clock later than rising edge of read strobe
15–13 IDCNT	Number of bytes that are read for the read id command.
12–8 TIMEOUT	The number of flash_clk cycles from NFC_WE high to either: • NAND flash busy (t_{WB}), or • NFC_RE low (t_{WHR}) After the last command is issued to flash, before sampling NFC_R/ $\overline{B}$, the NFC must wait t_{WB} clocks. After t_{WB} clocks: • if NFC_R/ $\overline{B}$ is sampled as high, the NFC considers the command to be a timeout, and the flash memory is idle. The NFC can issue new commands to the flash memory. • if NFC_R/ $\overline{B}$ is sampled as low, the NAND flash memory is busy. When reading the status or ID from the NAND flash memory, after the last command is issued to flash, the NFC must wait for t_{WHR} cycles. The NFC then negates NFC_RE to low to read the valid status or ID. **NOTE:** t_{WB} exists in page program/read, block erase, etc. Refer to the NAND flash datasheet for details of t_{WB} and t_{WHR} .
7 BITWIDTH	Flash mode width 0 8-bit wide flash mode 1 16-bit wide flash mode

Table continues on the next page...

NFC_CFG field descriptions (continued)

Field	Description
6 BTMD	**NOTE:** Resets to 0 if no boot is performed from the NFC, 1 if NFC boot is performed 0 Normal mode 1 Boot mode
5 AIAD	Auto-incrementing of flash row address 0 Do not auto-increment flash row address 1 Auto-increment flash row address
4 AIBN	Auto-incrementing of buffer numbers 0 Do not auto-increment buffer number 1 Auto-increment buffer number
3–0 PAGECNT	Number of virtual pages (in one physical flash page) to be programmed or read, etc.

31.3.14 DMA channel 2 address (NFC_DMA_CH2)

Address: 400E_0000h base + 3F34h offset = 400E_3F34h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_DMA_CH2 field descriptions

Field	Description
31–0 ADDRESS	DMA channel 2 address, it is 8-byte aligned.

31.3.15 Interrupt status (NFC_ISR)

Address: 400E_0000h base + 3F38h offset = 400E_3F38h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WERR	DONE	IDLE	0	WERRNS	CMDBUSY	RESBUSY	ECCBUSY	DMABUSY	WERREN	DONEEN	IDLEEN	WERRCLR	DONECLR	IDLECLR	0
W													w1c	w1c	w1c	
Reset	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										RESBN		ECCBN		DMABN	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

NFC_ISR field descriptions

Field	Description
31 WERR	Write error interrupt Set if an error condition is detected during a flash read status command. Sticky bit.
30 DONE	DONE interrupt Set if command processing is done.
29 IDLE	Command idle interrupt Set if the command is done, and residue engine, ECC engine and DMA engine are idle.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 WERRNS	Write error status Set if an error condition was detected during the last flash read status command. Non-sticky bit.

Table continues on the next page...

NFC_ISR field descriptions (continued)

Field	Description
26 CMDBUSY	Command busy Set it command execution busy, cleared otherwise.
25 RESBUSY	Residue engine busy Set if residue engine busy, cleared otherwise.
24 ECCBUSY	ECC engine busy Set if ECC engine busy, cleared otherwise.
23 DMABUSY	DMA engine busy Set if DMA engine busy, cleared otherwise.
22 WERREN	Enable bit for NFC_ISR[WERR]
21 DONEEN	Enable bit for NFC_ISR[DONE]
20 IDLEEN	Enable bit for NFC_ISR[IDLE]
19 WERRCLR	Clear bit for NFC_ISR[WERR]. Writing 1 to this bit clears NFC_ISR[WERR].
18 DONECLR	Clear bit for NFC_ISR[DONE]. Writing 1 to this bit clears NFC_ISR[DONE].
17 IDLECLR	Clear bit for NFC_ISR[IDLE]. Writing 1 to this bit clears NFC_ISR[IDLE].
16–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 RESBN	Residue buffer number Buffer number corresponding with the current residue block task.
3–2 ECCBN	ECC buffer number Buffer number corresponding with the current ECC task.
1–0 DMABN	DMA buffer number Buffer number corresponding with the current DMA task.

31.4 Functional Description

The NFC executes commands on a single or bank of external NAND flash chips. The NFC supports commands such as read, program, reset, erase, status read, read ID.

The NFC block contains a DMA engine and built-in ECC logic. For each read or write, the NFC performs ECC calculations on-the-fly. Two DMA channels are organized for each read or write: one for the main area, and one for the spare area. It is possible to disable the second DMA channel, and transfer main and spare data with only the first DMA channel.

Page size supported is 512, 2K, 4K, 8K and 16K bytes. There are 8 different ECC settings provided: 0, 4, 6, 8, 12, 16, 24 and 32 bits errors. They use 0, 8, 12, 15, 23, 30, 45 and 60 ECC bytes. The ECC works on page sizes of 512+spares bytes, 1K+spares bytes, 2K+spares bytes. The ECC algorithm used is a BCH code.

The error corrector can write ECC status to the spare area, since the read is pipelined. This means, while the current page is transferred from flash to buffer, the previous page is ECC corrected, and the page before that is transferred using DMA. Because of the pipelining, it is difficult to inform the CPU in the foreground of ECC errors. To solve this, ECC status is written to the auxiliary area of the sector, and transferred to memory. See Error Corrector Status for more information. It's up to the CPU to inspect the ECC result in memory, and act appropriately.

As described, reads are pipelined. However, writes are flow-through; no advance operations are done during a write. If there is a problem found during a write, the command sequence may be interrupted, and the CPU is informed.

Each page read, page write, page erase, read ID, or read status command sequence needs CPU attention only once. The CPU needs to prepare the DMA to point to the data, write correct values to all registers, and start the command. After command completion, the NFC block may interrupt the CPU.

The block allows command repeat, which is useful for write, read and erase, and allows processing multiple pages with just one command given by the CPU. No bank interleaving is supported during command repeat.

Booting from NAND flash is optional, and how it is activated is device-specific. See the Chip Configuration details for how to activate NFC boot on your device.

If boot feature is activated, first the NFC issues a reset command (0xFF) to the flash, then NFC reads four pages from block 0. Each page is 1056 bytes. The boot pages are protected by 32-bit error correction, which means that of the 1056 bytes, 996 bytes are user bytes and 60 bytes are ECC bytes. When the data from the boot pages is read, successfully error corrected, and stored in the NFC SRAM, the NFC indicates to the CPU that its boot code is visible in the NFC SRAM, and visible on addresses 0x000 to 0xF8F (3984 bytes total).

If the boot image from block 0 cannot be corrected, because there are more than 32 errors in one or several pages, boot is retried on the blocks at row addresses 256, 512, and 768. If it still fails after these retries, boot from the NFC is aborted. The device may begin execution using other memory. See the relevant Chip Configuration section for details on how your device behaves in this scenario.

Right after boot, a special address hashing function is active on all reads and writes DONE to NFC SRAM. This hashing function interleaves the page data from the four boot pages in such a way that all user data is visible in address range 0x000 to 0xF8F instead of four different ranges, one for each page. This hashing is controlled by NFC_CFG[BTMD], and the hashing should be turned off by the CPU after finishing reading/executing the boot image, and before normal operations of the NFC. See Figure 31-25 and NFC Buffer Memory Space.

Page size at boot is set to 1056 bytes to be compatible with a large number of NFC devices, without needing additional power-on reset flags to indicate the boot device.

- Compatible with 8-wide SLC and MLC devices with page size of 2 KB + 64 bytes spare
- Compatible with 8-wide SLC and MLC devices with page size of 4 KB and larger
- Compatible with 16-wide SLC and MLC devices with page size of 2 KB + 64 bytes spare
- Compatible with 16-wide SLC and MLC devices with page size of 4 KB and larger
- Not compatible with devices with 512 bytes page size.

31.4.1 NFC Buffer Memory Space

The next figure shows the organization of the buffer memory space in the NFC. The memory's size is 1152 × 64 bit, and is separated into four buffers, each with incontinuous physical address. For example, buffer 0's physical address is (0x000 + 0x20×i) – (0x007+0x20×i).

However, when the CPU writes or reads a buffer in non-boot mode, the CPU address is continuous, since there's an address transition inside NFC: sram_physical_addr[13:3] = {cpu_addr[11:3],cpu_addr[13:12]}

So, in non-boot mode, the address ranges are:

- Buffer 0: 0x0000 – 0x08FF
- Buffer 1: 0x1000 – 0x18FF

- Buffer 2: 0x2000 – 0x28FF
- Buffer 3: 0x3000 – 0x38FF

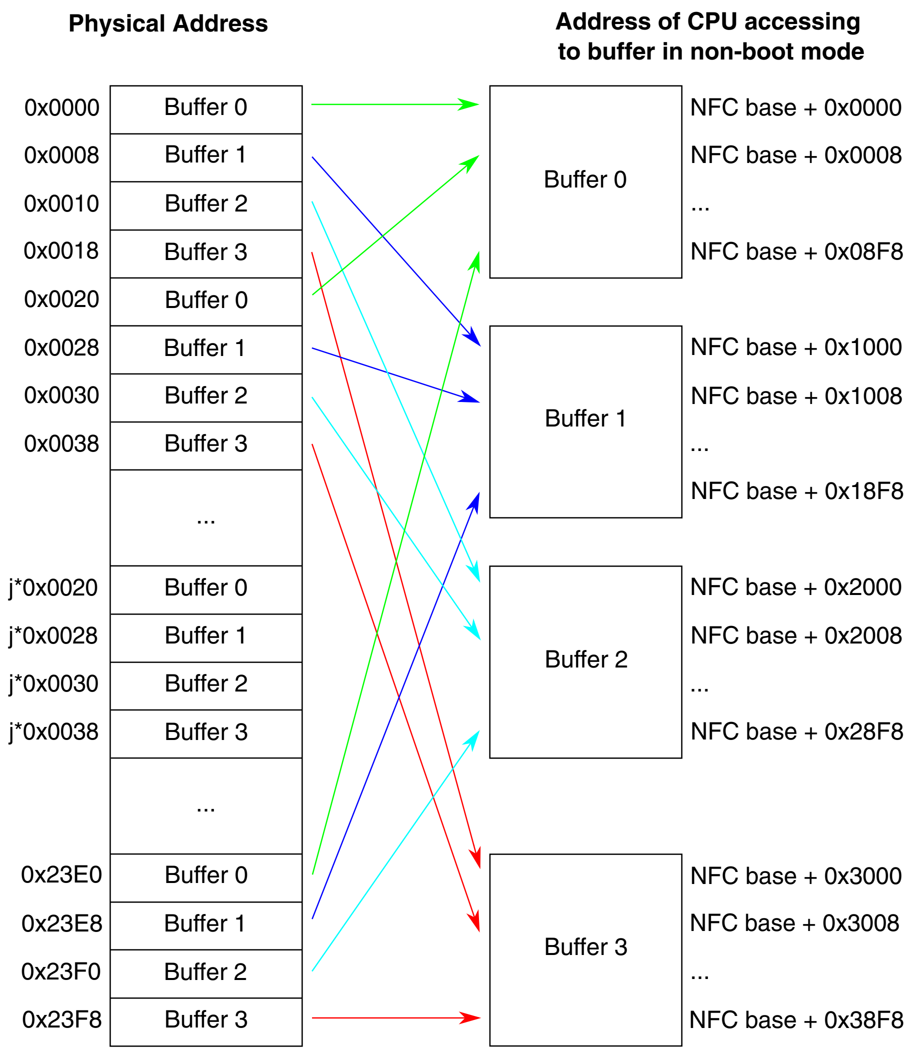


Figure 31-17. NFC Buffer Memory Space

31.4.2 Error Corrector Status

The ECC engine determines if a page is correctable. If correctable, it corrects error bits, and indicates error number. Otherwise, CORFAIL bit is asserted as shown in the next table. For a bad block management strategy to work, it may be necessary for the processor to obtain this information.

The error corrector writes the status word to a byte location to the SRAM buffer, defined by NFC_CFG[ECCAD]. It is selectable if the status is written or not with NFC_CFG[ECCSRAM]. If the status is written to the SRAM buffer, it becomes

effectively part of the flash data, and is processed like the flash data. Most likely, the status byte is written to memory as part of the page header. Once in memory, the ECC status is visible to the CPU, while CPU parses the rest of the flash header. No interrupt on error or status is available because this increases the interrupt load on the CPU. (The interrupt would be independent of the command done interrupt.) It is not possible to stop reading when ECC fails.

The organization of the status byte is shown here.

Table 31-18. ECC Status Word

Field	Definition
7 CORFAIL	0 Page has been successfully corrected 1 Page is uncorrectable
5–0 ERROR_COUNT	Number of errors that have been corrected in this page

NOTE

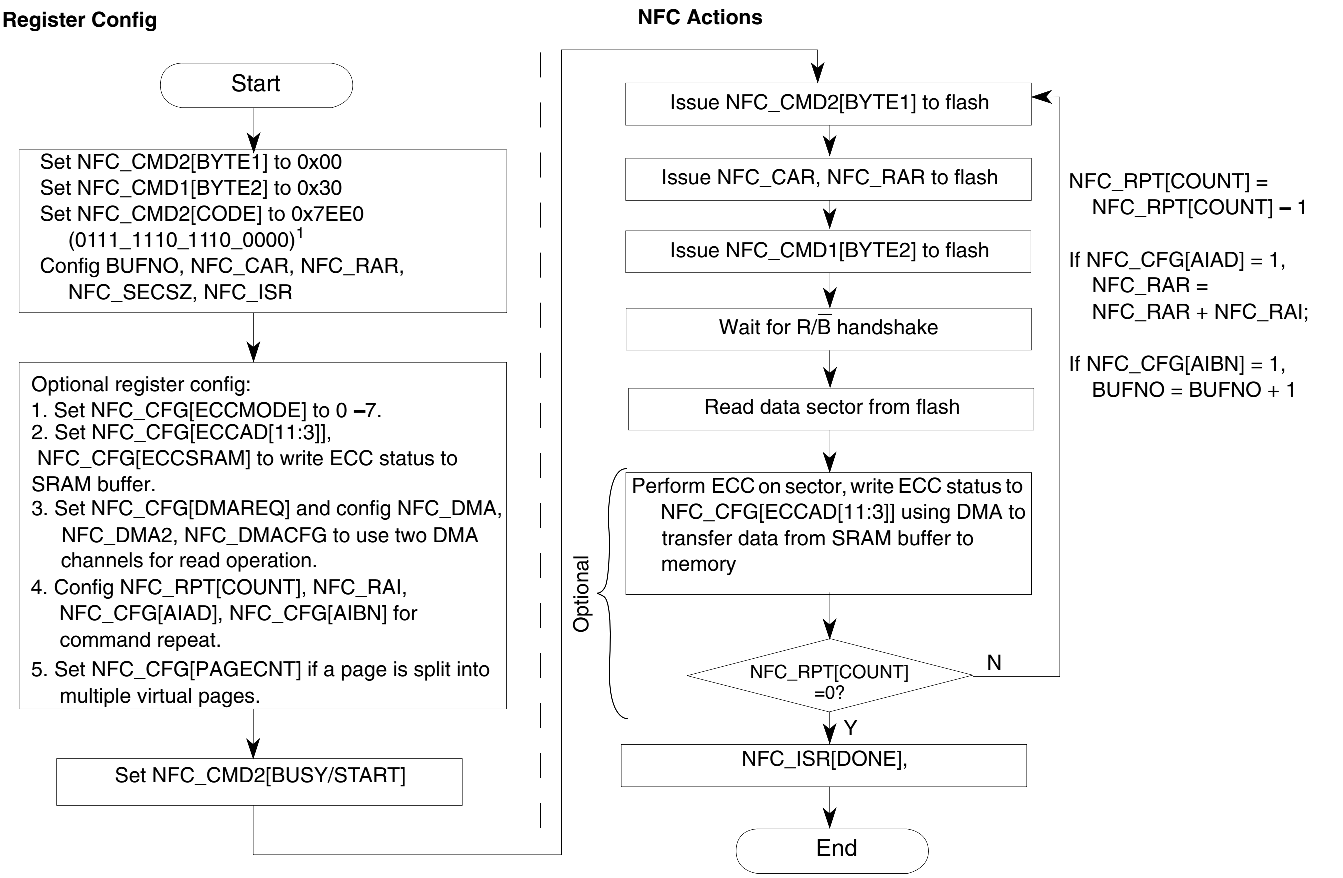
The address of the ECC status byte = Buffer n's start address + ECCAD[11:0] + 7 (n=0,1,2,3).

31.4.3 NFC Basic Commands

NFC basic commands include Page Read, Page Program, Block Erase, Read ID, and Reset.

31.4.3.1 Page Read

This command reads pages from the NAND flash. This figure shows the general flowchart of a read operation.



Note:
1 COL_ADDR2, NFC_RAR[BYTE3], and NFC_CMD1[BYTE2] (bold) are not necessary for some flash devices. See their data sheets for detail. For example:

If the flash only has one column address, then NFC_CMD2[CODE] = 0110_1110_1110_0000;
If the flash only has two row addresses, then NFC_CMD2[CODE] = 0111_1100_1110_0000;
If flash does not need the second command 0x30 for read, then NFC_CMD2[CODE] = 0110_1110_0110_0000.

Figure 31-18. Flow Chart of Read Operation

The next figure shows a particular case: one page is split into 8 virtual pages (see Organization of the Data in the NAND Flash), and DMA is not used. The SRAM buffer can hold data for four (virtual) pages at most. The CPU must transfer data out of the SRAM buffer after the first four virtual pages are read from flash. Otherwise, the next four virtual pages data overwrite the buffer. So, the read operation has following steps:

- Configure registers as shown in the preceding figure. NFC_CFG[PAGECNT] = 4, start commands, wait for NFC_ISR[DONE]
- CPU reads data from buffer, set NFC_CMD2[CODE] = 0x20 (only enable read data)
- Start commands to read out the next 4 virtual pages, wait for NFC_ISR[DONE]

If DMA is used to transfer data from SRAM buffer to memory instead of CPU, the flow in the preceding figure is used: set NFC_CFG[PAGECNT] = 8, set NFC_CFG[DMAREQ] = 1, configure DMA registers, start commands. A pipeline (Functional Description) controls the read operation.

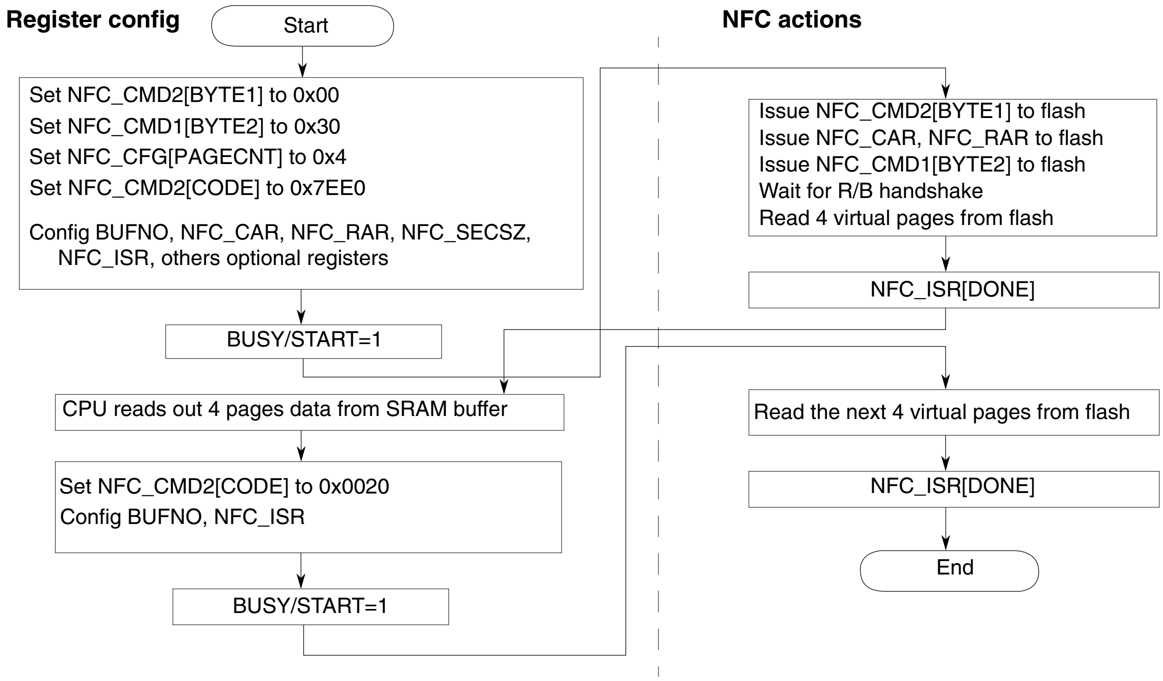


Figure 31-19. Flow Chart of Read Operation, NFC_CFG[PAGECNT] = 8, No DMA

Note

See footnote in Figure 31-18.

31.4.3.2 Page Program

This command programs pages to the NAND flash. The next figure is the general flow of page program operation.

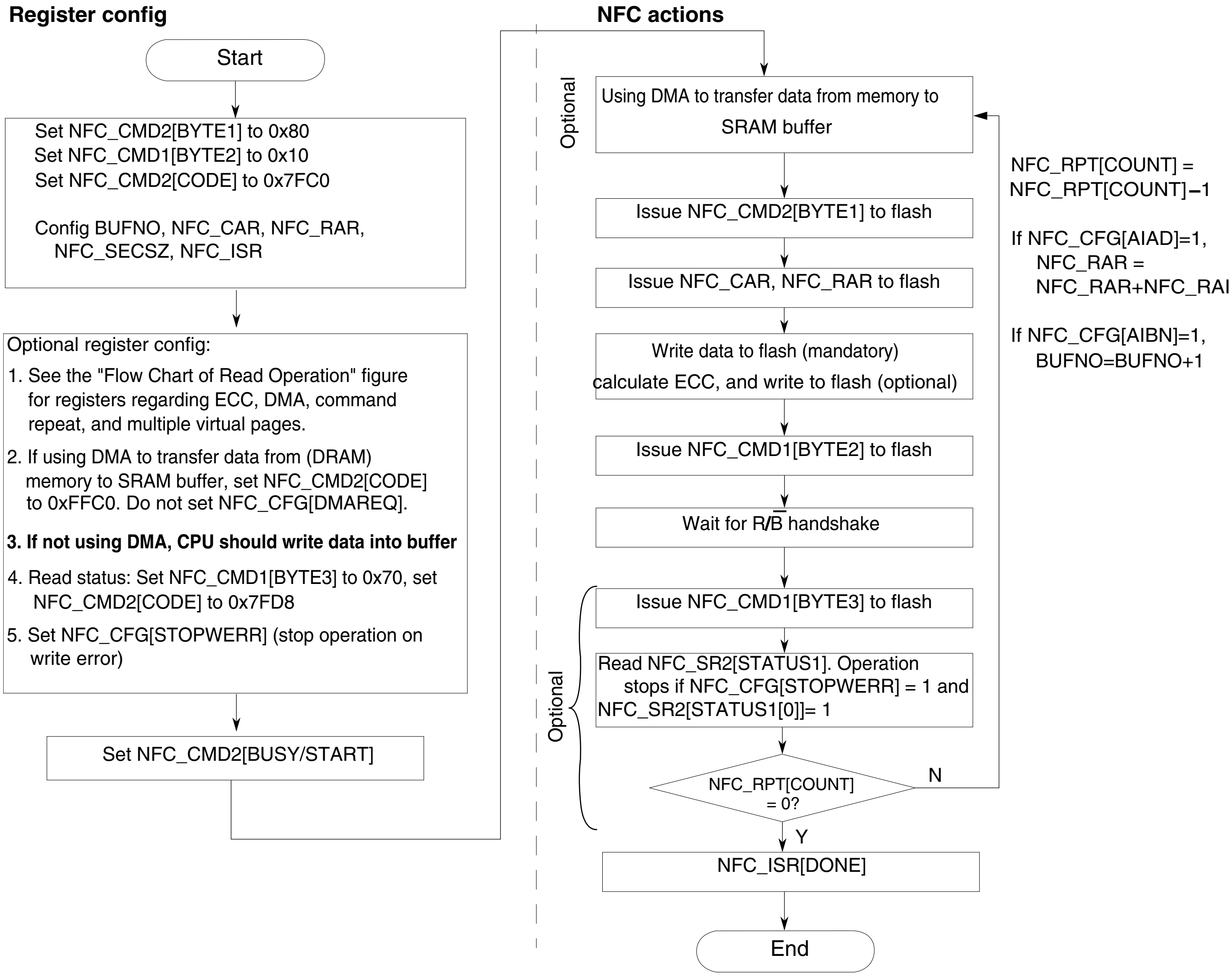


Figure 31-20. Flow Chart of Page Program Operation

Note

COL_ADDR2 and NFC_RAR[BYTE3] (bold) are not necessary for some flash devices. See their data sheets for detail. See the footnote of Figure 31-18.

The next figure shows the particular case which is similar to Figure 31-19. The CPU writes at most four virtual pages of data into the buffer before the first start command. Set NFC_CFG[PAGECNT] to 4 and set NFC_CMD2[CODE] twice:

- First, set it to 0x7F00 (0111_1111_0000_0000). The NFC issues NFC_CMD2[BYTE1], address cycles, four virtual pages data to flash. After NFC_ISR[DONE] is set, the CPU can write the next four virtual pages data into the SRAM buffer.
- Second, set CODE to 0x01C0 (0000_0001_1100_0000). The NFC sends the next four virtual pages of data to flash, issues NFC_CMD1[BYTE2], waits for R/$\overline{B}$ handshake, and waits for NFC_ISR[DONE] to set.

Like the read operation, if DMA transfers data from memory to NFC SRAM buffer (instead of the CPU), the flow in the preceding figure is used and set NFC_CFG[PAGECNT] to 0x8.

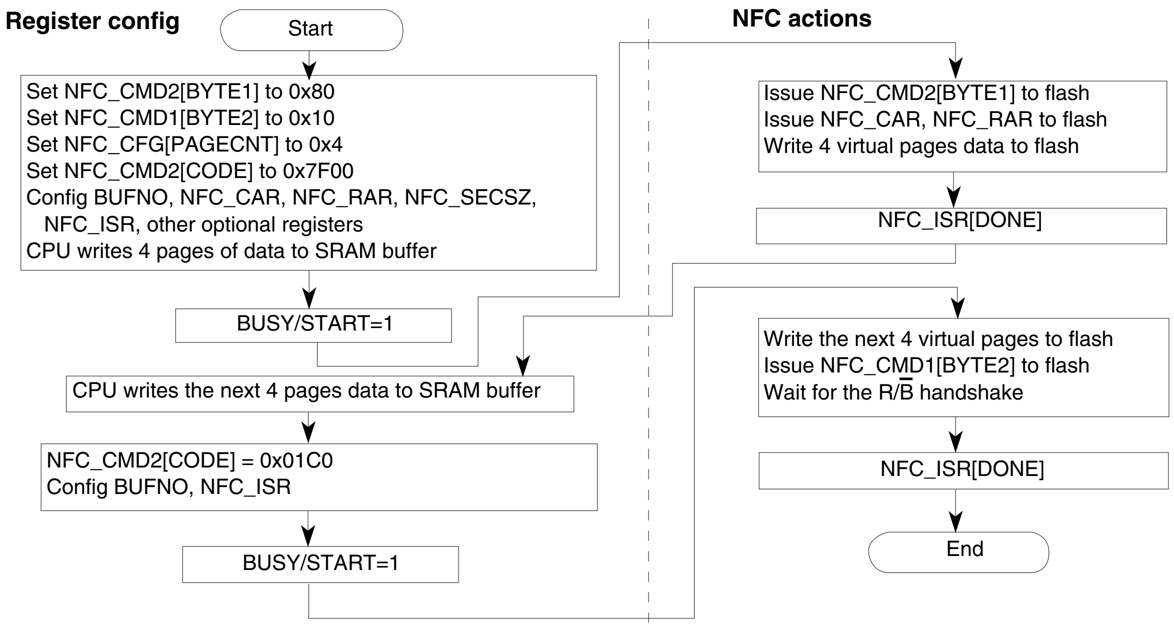


Figure 31-21. Flow Chart of Page Program Operation, NFC_CFG[PAGECNT] = 8, No DMA

Note

If you want to read the status after the second 0x10 command, set NFC_CMD1[BYTE3] to 0x70 and NFC_CMD2[CODE] to 0x01D8 (0000_0001_1101_1000). Then, after "Wait for the R/$\overline{B}$ handshake", the NFC issues NFC_CMD1[BYTE3] to flash, and reads the status. If NFC_CFG[STOPWERR] is set and NFC_SR2[STATUS1[0]]=1, operation stops. Otherwise, NFC_ISR[DONE] comes out. The COL_ADDR2 and NFC_RAR[BYTE3] of the first NFC_CMD2[CODE] may not be necessary. See note 1 of Figure 31-18.

31.4.3.3 Block Erase

This command is used to erase blocks.

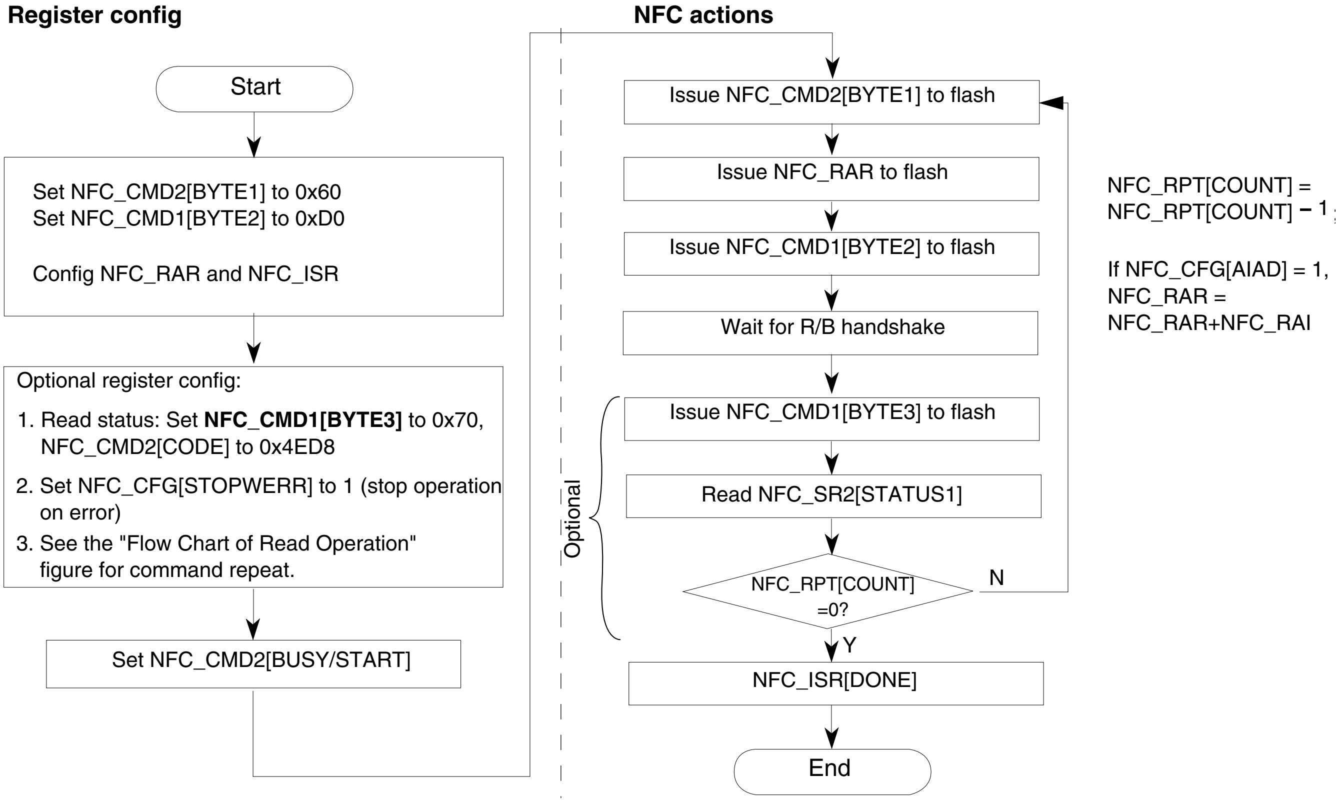


Figure 31-22. Flow Chart of Block Erase Operation

Note

NFC_RAR[BYTE3] (bold) is not necessary for some flash devices. See their data sheets for detail.

Note

If NFC_CFG[STOPWERR] is set and NFC_SR2[STATUS1[0]]=1, operation stops.

31.4.3.4 Read ID

This command reads the flash ID.

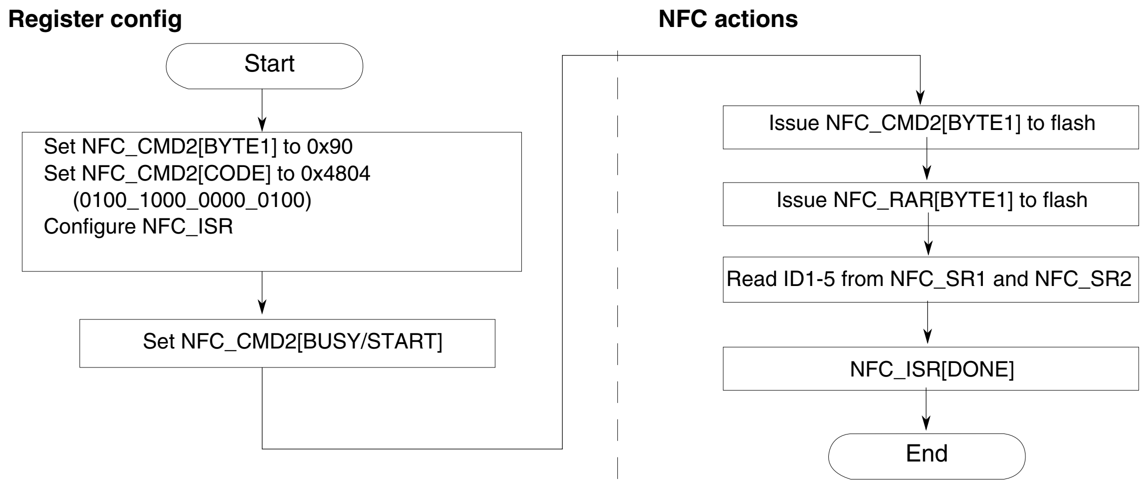


Figure 31-23. Flow Chart of Read ID Operation

31.4.3.5 Reset

This command sends a single reset command to the flash.

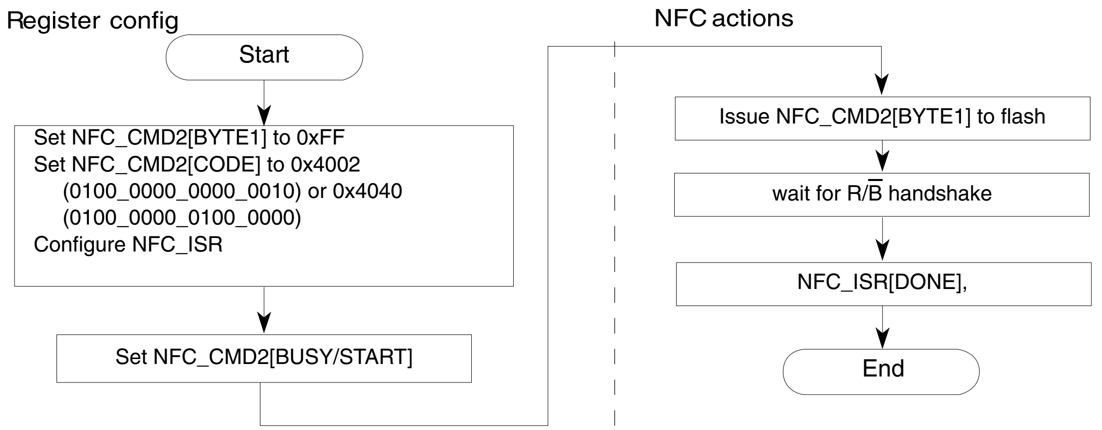


Figure 31-24. Flow Chart of Reset Operation

31.4.4 NAND Flash Boot

For booting, the power-on reset values of some registers are:

- Sector size is 1056 bytes: NFC_SECSZ = 1056
- Flash is defined as 8-bit: NFC_CFG[16BIT] = 0
- ECC correction depth is 32 bits errors: NFC_CFG[ECCMODE] = 0x7
- Boot sector address is 0: NFC_CMD2[BUFNO] = 00

Boot-up occurs after power-on reset. After boot, the following happens:

1. The NFC issues reset command 0xFF to the flash.
2. Four boot blocks are identified in the flash at row addresses 0, 256, 512, and 768.
3. The flash controller burst reads four pages from the first boot block to memory. A total of 4 KB are read in this way.
4. If there is no ECC failure during the burst read, the boot is successful.

   If there is an ECC failure during the burst read, the process is started again from the next boot block. If the fourth block still has ECC failure, boot is unsuccessful, which means it is not possible to read a reliable boot image from the flash.
5. CPU access is held until boot completion.
6. After boot, the boot image is visible in the memory map at address 0x00 – 0xF8F. A special hash function is active on the NFC SRAM read to have the boot image in one continuous address range, and not in four address ranges (one for each page). The NFC_CFG[BTMD] bit controls this hash function, and this bit should be cleared after the CPU has read/executed the boot image, and before it operates the NFC in standard mode. See the next figure.

Note

There is no difference in how data is transferred between SRAM buffer and the flash: one page uses one buffer. But, how the CPU writes/reads the buffer is different. In non-boot mode, the write/read address is the SRAM physical address. In boot mode, the write/read address is based on the buffers. See NFC Buffer Memory Space.

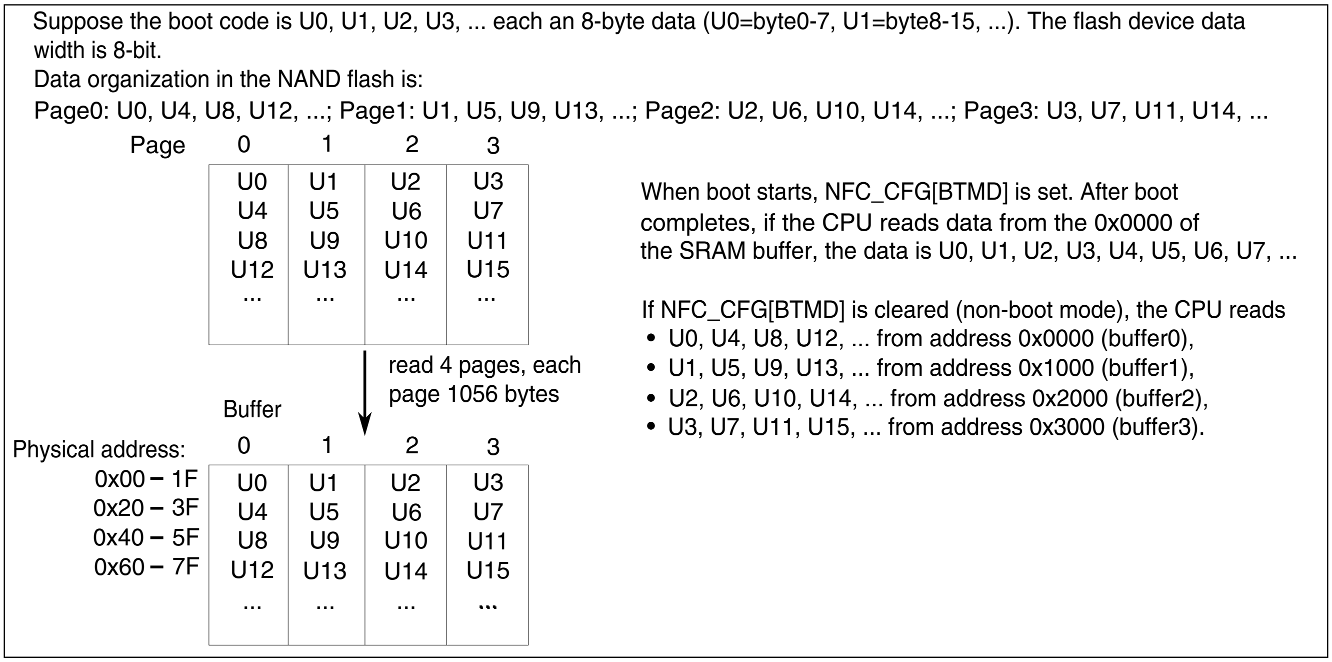


Figure 31-25. Boot and NFC_CFG[BTMD]

The CPU boot code should be split into four pages as shown in the preceding figure. Boot sector encoding cannot be done by the NFC block itself. It must be done in software. Each page contains 996-byte boot codes and 60-byte ECC codes.

The ECC codes should be inverted because when NFC is doing a page program, it sends the inverted ECC codes to flash. When reading, it inverts them again to get the correct ECC codes for error correction. And only page reads are performed in boot operation. So, software should do the ECC code inversion after boot sector encoding.

The boot is identical for a 8- and 16-bit wide flash. If a 16-bit flash is connected, the data on the upper lane is discarded. If a 8-bit flash is connected, the upper 1 KB of data is not read.

For an 8-bit flash, the upper 1 KB of data is not read.

Table 31-19. Boot Data's Location in a Page — 8-bit

8-bit flash boot codes and ECC's location in apage		
0		1056
996 bytes boot codes	60 bytes ECC (inverted)	Unused

Table 31-20. Boot Data's Location in a Page — 16-bit

	15	8 7	0
0x0000	xx	Bootcode byte 0	
0x0002	xx	Bootcode byte 1	

Table continues on the next page...

Table 31-20. Boot Data's Location in a Page — 16-bit (continued)

0x0004 ... 0x07C4	... xx ...	... Bootcode byte i ...
0x07C6	xx	Bootcode byte 995
0x07C8	xx	ECC byte 0 (inverted)
0x07CA	xx	ECC byte 1 (inverted)
0x07CC ... 0x083C	... xx ...	... ECC byte j (inverted) ...
0x083E	xx	ECC byte 59 (inverted)

31.4.5 Fast Flash Configuration for EDO

Normally, read out data goes valid after the high-to-low transition of $\overline{RE}$, and invalid on the low-to-high transition (as shown in the next figure) $t_{RHOH}<t_{REH}$. NFC sampled the read data at the negedge of flash_clk, and because the data is invalid at that time, a latch is used here to maintain the valid data during the high period of flash_clk, so that NFC can sample correct data.

Some flash devices contain an EDO (enhanced data out) feature, where the data can be held until the next high-to-low $\overline{RE}$ transition (see the figure after next, labeled "Read Operation, EDO type"), $t_{RHOH}>t_{REH}$. The read data is valid at the negedge of flash_clk, NFC can sample data directly without latching it. To support the EDO feature, the NFC must work in fast mode (NFC_CFG[FAST] set). The NFC clock must be configured fast enough (usually > 33 MHz) according to the data sheet of flash devices.

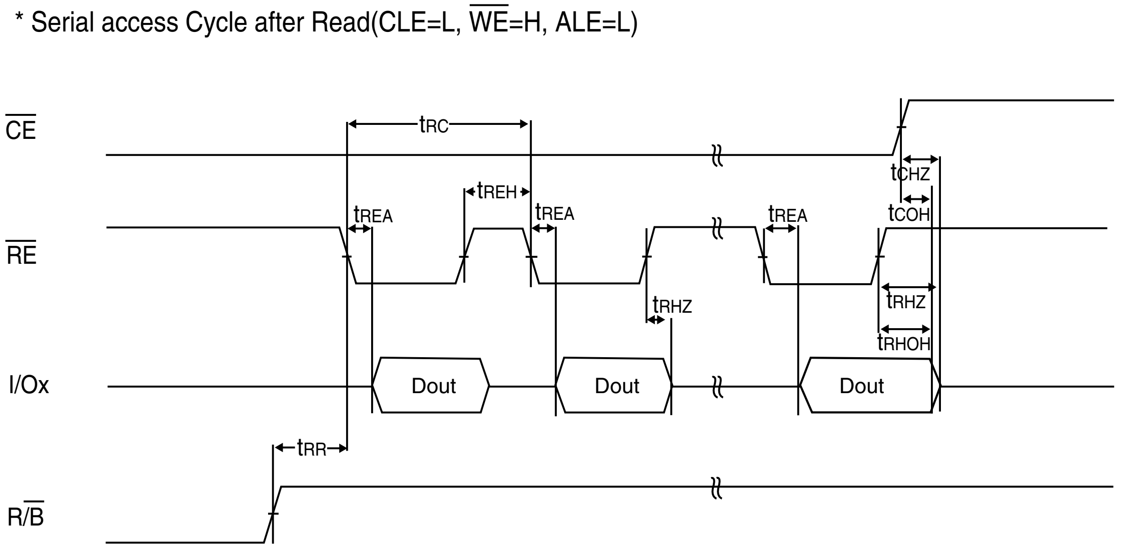


Figure 31-26. Read Operation

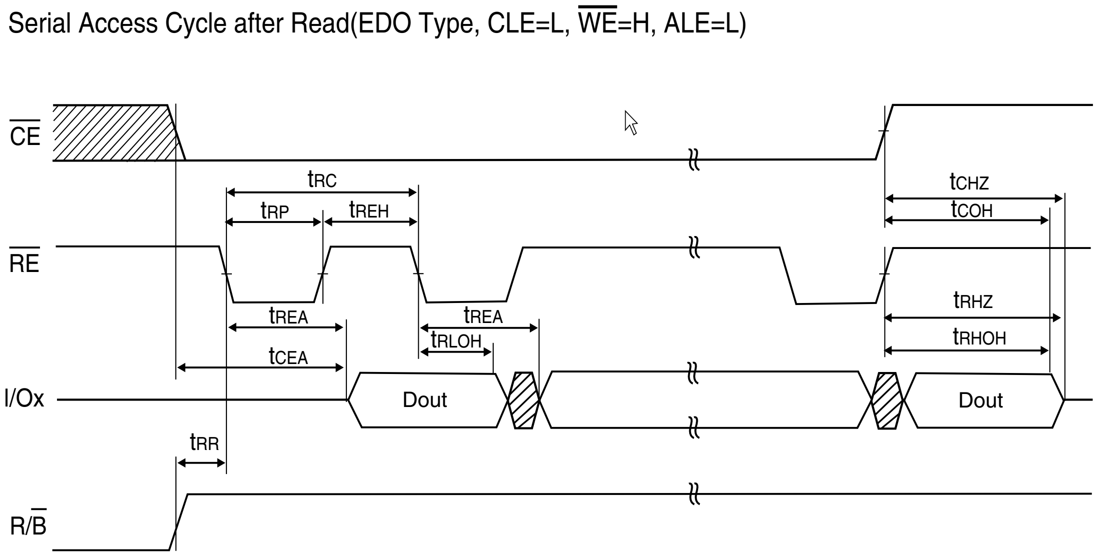


Figure 31-27. Read Operation, EDO type

31.4.6 Organization of the Data in the NAND Flash

Pages on the flash can be split into multiple virtual ECC/DMA pages. The parameter that controls this is NFC_CFG[PAGECNT]. This parameter gives the number of virtual ECC/DMA pages in one flash page.

The virtual page is split into a user (main) area and ECC (spare) area. Data in the user area can be set or used by the application, while data in the ECC area is set and used by the ECC.

The following tables give virtual-to-physical mappings for various flash devices and their recommended settings.

Table 31-21. Virtual-to-Physical Mappings of Different Flash,

Flash page size (main +spare) bytes	NFC_CFG [ECC MODE]	ECC bits	NFC_CFG [PAGE CNT]	Sector size (bytes)	Virtual page user size (bytes)	Mapping
512 + 16	000	0	1	528	528	VirtualPage_0[527:0] = Physical[527:0]
512 + 16	001	4	1	528	520	VirtualPage_0[519:0] = Physical[519:0]
2048 + 64	000	0	1	2112	2112	VirtualPage_0[2111:0] = Physical[2111:0]
2048 + 64	101	16	1	2112	2082	VirtualPage_0[2081:0] = Physical[2081:0]
2048 + 64	110	24	1	2112	2067	VirtualPage_0[2066:0] = Physical[2066:0]
2048 + 64	111	32	1	2112	2052	VirtualPage_0[2051:0] = Physical[2051:0]
2048 + 64	000	0	4	528	528	VirtualPage_0[527:0] = Physical[527:0] VirtualPage_1[527:0] = Physical[1055:528] VirtualPage_2[527:0] = Physical[1583:1056] VirtualPage_3[527:0] = Physical[2111:1584]
2048 + 64	001	4	4	528	520	VirtualPage_0[519:0] = Physical[519:0] VirtualPage_1[519:0] = Physical[1047:528] VirtualPage_2[519:0] = Physical[1575:1056] VirtualPage_3[519:0] = Physical[2103:1584]
4096 + 128	000	0	2	2112	2112	VirtualPage_0[2111:0] = Physical[2111:0] VirtualPage_1[2111:0] = Physical[4223:2112]
4096 + 128	101	16	2	2112	2082	VirtualPage_0[2081:0] = Physical[2081:0] VirtualPage_1[2081:0] = Physical[4193:2112]
4096 + 128	110	24	2	2112	2067	VirtualPage_0[2066:0] = Physical[2066:0] VirtualPage_1[2066:0] = Physical[4178:2112]
4096 + 128	111	32	2	2112	2052	VirtualPage_0[2051:0] = Physical[2051:0][1] VirtualPage_1[2051:0] = Physical[4163:2112]
4096 + 128	000	0	8[2]	528	528	VirtualPage_0[527:0] = Physical[527:0] VirtualPage_1[527:0] = Physical[1055:528] VirtualPage_2[527:0] = Physical[1583:1056] VirtualPage_3[527:0] = Physical[2111:1584] VirtualPage_4[527:0] = Physical[2639:2112] VirtualPage_5[527:0] = Physical[3167:2640] VirtualPage_6[527:0] = Physical[3695:3168] VirtualPage_7[527:0] = Physical[4223:3696]

Table continues on the next page...

Table 31-21. Virtual-to-Physical Mappings of Different Flash, (continued)

Flash page size (main +spare) bytes	NFC_CFG [ECC MODE]	ECC bits	NFC_CFG [PAGE CNT]	Sector size (bytes)	Virtual page user size (bytes)	Mapping
4096 + 128	001	4	8	528	520	VirtualPage_0[519:0] = Physical[519:0] VirtualPage_1[519:0] = Physical[1047:528] VirtualPage_2[519:0] = Physical[1575:1056] VirtualPage_3[519:0] = Physical[2103:1584] VirtualPage_4[519:0] = Physical[2631:2112] VirtualPage_5[519:0] = Physical[3159:2640] VirtualPage_6[519:0] = Physical[3687:3168] VirtualPage_7[519:0] = Physical[4215:3696]
4096 + 208	000	0	2	2152	2152	VirtualPage_0[2151:0] = Physical[2151:0] VirtualPage_1[2151:0] = Physical4303:2152]
4096 + 208	101	16	2	2152	2122	VirtualPage_0[2121:0] = Physical[2121:0] VirtualPage_1[2121:0] = Physical[4273:2152]
4096 + 208	110	24	2	2152	2104	VirtualPage_0[2103:0] = Physical[2103:0] VirtualPage_1[2103:0] = Physical[4255:2152]
4096 + 208	111	32	2	2152	2092	VirtualPage_0[2091:0] = Physical[2091:0] VirtualPage_1[2091:0] = Physical[4243:2152]
4096 + 208	000	0	8	538	538	VirtualPage_0[537:0] = Physical[537:0] VirtualPage_1[537:0] = Physical[1075:538] VirtualPage_2[537:0] = Physical[1613:1076] VirtualPage_3[537:0] = Physical[2151:1614] VirtualPage_4[537:0] = Physical[2689:2152] VirtualPage_5[537:0] = Physical[3227:2690] VirtualPage_6[537:0] = Physical[3765:3228] VirtualPage_7[537:0] = Physical[4304:3766]
4096 + 208	001	4	8	538	530	VirtualPage_0[529:0] = Physical[529:0] VirtualPage_1[529:0] = Physical[1067:538] VirtualPage_2[529:0] = Physical[1605:1076] VirtualPage_3[529:0] = Physical[2143:1614] VirtualPage_4[529:0] = Physical[2681:2152] VirtualPage_5[529:0] = Physical[3219:2690] VirtualPage_6[529:0] = Physical[3757:3228] VirtualPage_7[529:0] = Physical[4295:3766]

Table continues on the next page...

Table 31-21. Virtual-to-Physical Mappings of Different Flash, (continued)

Flash page size (main +spare) bytes	NFC_CFG [ECC MODE]	ECC bits	NFC_CFG [PAGE CNT]	Sector size (bytes)	Virtual page user size (bytes)	Mapping
4096 + 208	010	6	8	538	526	VirtualPage_0[525:0] = Physical[525:0]
						VirtualPage_1[525:0] = Physical[1063:538]
						VirtualPage_2[525:0] = Physical[1601:1076]
						VirtualPage_3[525:0] = Physical[2139:1614]
						VirtualPage_4[525:0] = Physical[2677:2152]
						VirtualPage_5[525:0] = Physical[3215:2690]
						VirtualPage_6[525:0] = Physical[3753:3228]
						VirtualPage_7[525:0] = Physical[4291:3766]
4096 + 208	011	8	8	538	523	VirtualPage_0[522:0] = Physical[523:0]
						VirtualPage_1[522:0] = Physical[1060:538]
						VirtualPage_2[522:0] = Physical[1598:1076]
						VirtualPage_3[522:0] = Physical[2136:1614]
						VirtualPage_4[522:0] = Physical[2674:2152]
						VirtualPage_5[522:0] = Physical[3212:2690]
						VirtualPage_6[522:0] = Physical[3750:3228]
						VirtualPage_7[522:0] = Physical[4288:3766]

1. In most applications, this mode is of no use because user size is too small.
2. When 4KB page is split into eight virtual pages, if page program/read using DMA, set NFC_CFG[PAGECNT] to 8. if not using DMA, set NFC_CFG[PAGECNT] to 4. See Page Read and Page Program for details.

If flash devices with a physical page size of 4K or more are used, the bad block marker appears as the first byte of the spare area. But, because of the physical-to-virtual mapping, it does not appear in byte 2048 of the virtual page, where its logical place would be. The DMA engine contains the option to swap some bytes, and to make the bad block marker appear in the requested place.

Table 31-22. Using the Swap Field to Move the Bad Block Marker

Flash sector size (main + spare) bytes	Bad block marker (physical)	Bad block marker (virtual) Before swap	Bad block marker (expected) After swap	Swap
4096 + 128	4096	Page 1/byte 1984	Page 1/byte 2048	NFC_SWAP[ADDR1] = (1984/8)[1]
				NFC_SWAP[ADDR2] = (2048/8)
4096 + 208	4096	Page 1/byte 1944	Page 1/byte 2048	NFC_SWAP[ADDR1] = (1944/8)
				NFC_SWAP[ADDR2] = (2048/8)

1. Only works with a user page size of at least 2055 bytes. Does not work with ECC mode 111.

31.4.7 Flash Command Code Description

The 16-bit command code in NFC_CMD2[CODE] is defined in the next table. If a bit is set, the action is executed. The command is repeated for the number of the NFC_RPT[COUNT] value. If NFC_RPT[COUNT] is zero or one, the command is executed once.

Table 31-23. NFC_CMD2[CODE] Detail

NFC_CMD2 [CODE] bit	Action when Bit is Set
15	Start DMA transfer to read data from memory , and write to SRAM.
14	Send command byte 1 (NFC_CMD2[BYTE1]) to flash
13	Send column address 1 (NFC_CAR[BYTE1]) to flash
12	Send column address 2 (NFC_CAR[BYTE2]) to flash
11	Send row address 1 (NFC_RAR[BYTE1]) to flash
10	Send row address 2 (NFC_RAR[BYTE2]) to flash
9	Send row address 3 (NFC_RAR[BYTE3]) to flash
8	Write data to flash. Total of NFC_CFG[PAGECNT] pages is written to the flash, and equal number of starts is sent to the residue engine. Also, additional starts to the DMA engine are sent, until DMA has transferred the NFC_CFG[PAGECNT] data from memory to NFC.
7	Send command byte 2 (NFC_CMD1[BYTE2]) to flash
6	Wait for flash R/$\overline{B}$ handshake
5	Read data from flash. Read is only started if the new NFC_CMD2[BUFNO] is idle. One or more starts are sent to the residue engine, total NFC_CFG[PAGECNT] starts. **Note:** For reads, DMA is not started. Instead, to start DMA for reads, NFC_CFG[DMAREQ] must be set.
4	Send command byte 3 (NFC_CMD1[BYTE3]) to flash
3	Read flash status
2	Read ID
1	Always set. End-of-command marker used to signal done.
0	Reserved, must be cleared.

31.4.8 Interrupts

There are two interrupts to flag the end of a command execution:

1. The DONE interrupt, NFC_ISR[DONE]. Use this interrupt if commands are sent back-to-back to the flash. It indicates when a new command can be dispatched. The DONE interrupt is given before the flash data is corrected and resident in memory, because the operation of the ECC engine and DMA engine is pipelined.

When the DONE interrupt tracks command completion, the software may also monitor the NFC_ISR[ECCBUSY, DMABUSY, ECCBN, DMABN] fields.

a. NFC_ISR[ECCBUSY] indicates that the ECC block is still busy, and reports the buffer number the ECC block is working on in NFC_ISR[ECCBN].
b. NFC_ISR[DMABUSY] indicates that the DMA block is still busy, and reports the buffer number the DMA block is working on in NFC_ISR[DMABN].

2. The command IDLE interrupt, NFC_ISR[IDLE]. Use this interrupt if you want to use the data produced in the next process. The IDLE interrupt indicates all command processing has terminated, and the relevant data is now available in memory or the NFC SRAM buffer. When using back-to-back reads to the flash, use of the IDLE interrupt means the NFC does not operate at its maximum transfer speed, as ECC and DMA are now done in the foreground.

When using the DONE interrupt, transfer completion for write pages can be assumed when the DONE interrupt is received. When DONE is received for read pages, the data may still be in flight in the DMA or the ECC. To check this, the CPU should remember the buffer number (NFC_CMD2[BUFNO]) associated with the command, and wait until the DMA and ECC are either idle, or are both busy on a different buffer number. (The ECC buffer number and the DMA buffer number fields do not match the BUFNO specified with the command.) You can check on any DONE interrupt or by polling the register.

Chapter 32
External Bus Interface (FlexBus)

32.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

This chapter describes external bus data transfer operations and error conditions. It describes transfers initiated by the core processor (or any other bus master) and includes detailed timing diagrams showing the interaction of signals in supported bus operations.

32.1.1 Definition

The FlexBus multifunction external bus interface controller is a hardware module that:

- Provides memory expansion and provides connection to external peripherals with a parallel bus
- Can be directly connected to the following asynchronous or synchronous slave-only devices with little or no additional circuitry:
  - External ROMs
  - Flash memories
  - Programmable logic devices
  - Other simple target (slave) devices

32.1.2 Features

FlexBus offers the following features:

- Five independent, user-programmable chip-select signals ($\overline{\text{FB_CS}}$4 –$\overline{\text{FB_CS}}$0)
- 8-bit, 16-bit, and 32-bit port sizes with configuration for multiplexed or nonmultiplexed address and data buses
- 8-bit, 16-bit, 32-bit, and 16-byte transfers
- Programmable burst and burst-inhibited transfers selectable for each chip-select and transfer direction
- Programmable address-setup time with respect to the assertion of a chip-select
- Programmable address-hold time with respect to the deassertion of a chip-select and transfer direction
- Extended address latch enable option to assist with glueless connections to synchronous and asynchronous memory devices

32.2 Signal descriptions

This table describes the external signals involved in data-transfer operations.

NOTE

Not all of the following signals may be available on a particular device. See the Chip Configuration details for information on which signals are available.

Table 32-1. FlexBus signal descriptions

Signal	I/O	Function
FB_A31–FB_A0	O	Address Bus When FlexBus is used in a nonmultiplexed configuration, this is the address bus. When FlexBus is used in a multiplexed configuration, this bus is not used.
FB_D31–FB_D0	I/O	Data Bus—During the first cycle, this bus drives the upper address byte, addr[31:24]. When FlexBus is used in a nonmultiplexed configuration, this is the data bus, FB_D. When FlexBus is used in a multiplexed configuration, this is the address and data bus, FB_AD. The number of byte lanes carrying the data is determined by the port size associated with the matching chip-select. When FlexBus is used in a multiplexed configuration, the full 32-bit address is driven on the first clock of a bus cycle (address phase). After the first clock, the data is driven on the bus (data phase). During the data phase, the address is driven on the pins not used for data. For example, in 16-bit mode, the lower address is driven on FB_AD15–FB_AD0, and in 8-bit mode, the lower address is driven on FB_AD23–FB_AD0.

Table continues on the next page...

Table 32-1. FlexBus signal descriptions (continued)

Signal	I/O	Function
FB_CS4–FB_CS0	O	General Purpose Chip-Selects—Indicate which external memory or peripheral is selected. A particular chip-select is asserted when the transfer address is within the external memory's or peripheral's address space, as defined in CSAR[BA] and CSMR[BAM].
FB_BE_31_24 FB_BE_23_16 FB_BE_15_8 FB_BE_7_0	O	Byte Enables—Indicate that data is to be latched or driven onto a specific byte lane of the data bus. CSCR[BEM] determines if these signals are asserted on reads and writes or on writes only. For external SRAM or flash devices, the FB_BE outputs should be connected to individual byte strobe signals.
FB_OE	O	Output Enable—Sent to the external memory or peripheral to enable a read transfer. This signal is asserted during read accesses only when a chip-select matches the current address decode.
FB_R/W	O	Read/Write—Indicates whether the current bus operation is a read operation (FB_R/W high) or a write operation (FB_R/W low).
FB_TS	O	Transfer Start—Indicates that the chip has begun a bus transaction and that the address and attributes are valid. An inverted FB_TS is available as an address latch enable (FB_ALE), which indicates when the address is being driven on the FB_AD bus. FB_TS/FB_ALE is asserted for one bus clock cycle. The chip can extend this signal until the first positive clock edge after FB_CS asserts. See CSCR[EXTS] and Extended Transfer Start/Address Latch Enable.
FB_ALE	O	Address Latch Enable—Indicates when the address is being driven on the FB_A bus (inverse of FB_TS).

Table continues on the next page...

Table 32-1. FlexBus signal descriptions (continued)

Signal	I/O	Function
FB_TSIZ1–FB_TSIZ0	O	Transfer Size—Indicates (along with FB_TBST) the data transfer size of the current bus operation. The interface supports 8-, 16-, and 32-bit operand transfers and allows accesses to 8-, 16-, and 32-bit data ports. • 00b = 4 bytes • 01b = 1 byte • 10b = 2 bytes • 11b = 16 bytes (line) For misaligned transfers, FB_TSIZ1–FB_TSIZ0 indicate the size of each transfer. For example, if a 32-bit access through a 32-bit port device occurs at a misaligned offset of 1h, 8 bits are transferred first (FB_TSIZ1–FB_TSIZ0 = 01b), 16 bits are transferred next at offset 2h (FB_TSIZ1–FB_TSIZ0 = 10b), and the final 8 bits are transferred at offset 4h (FB_TSIZ1–FB_TSIZ0 = 01b). For aligned transfers larger than the port size, FB_TSIZ1–FB_TSIZ0 behave as follows: • If bursting is used, FB_TSIZ1–FB_TSIZ0 are driven to the transfer size. • If bursting is inhibited, FB_TSIZ1–FB_TSIZ0 first show the entire transfer size and then show the port size. For burst-inhibited transfers, FB_TSIZ1–FB_TSIZ0 change with each FB_TS assertion to reflect the next transfer size. For transfers to port sizes smaller than the transfer size, FB_TSIZ1–FB_TSIZ0 indicate the size of the entire transfer on the first access and the size of the current port transfer on subsequent transfers. For example, for a 32-bit write to an 8-bit port, FB_TSIZ1–FB_TSIZ0 are 00b for the first transaction and 01b for the next three transactions. If bursting is used for a 32-bit write to an 8-bit port, FB_TSIZ1–FB_TSIZ0 are driven to 00b for the entire transfer.
FB_TBST	O	Transfer Burst—Indicates that a burst transfer is in progress as driven by the chip. A burst transfer can be 2 to 16 beats depending on FB_TSIZ1–FB_TSIZ0 and the port size. **Note:** When a burst transfer is in progress (FB_TBST = 0b), the transfer size is 16 bytes (FB_TSIZ1–FB_TSIZ0 = 11b), and the address is misaligned within the 16-byte boundary, the external memory or peripheral must be able to wrap around the address.

Table continues on the next page...

Table 32-1. FlexBus signal descriptions (continued)

Signal	I/O	Function
FB_TA	I	Transfer Acknowledge—Indicates that the external data transfer is complete. When FB_TA is asserted during a read transfer, FlexBus latches the data and then terminates the transfer. When FB_TA is asserted during a write transfer, the transfer is terminated. If auto-acknowledge is disabled (CSCR[AA] = 0), the external memory or peripheral drives FB_TA to terminate the transfer. If auto-acknowledge is enabled (CSCR[AA] = 1), FB_TA is generated internally after a specified number of wait states, or the external memory or peripheral may assert external FB_TA before the wait-state countdown to terminate the transfer early. The chip deasserts FB_CS one cycle after the last FB_TA is asserted. During read transfers, the external memory or peripheral must continue to drive data until FB_TA is recognized. For write transfers, the chip continues driving data one clock cycle after FB_CS is deasserted. The number of wait states is determined by CSCR or the external FB_TA input. If the external FB_TA is used, the external memory or peripheral has complete control of the number of wait states. **Note:** External memory or peripherals should assert FB_TA only while the FB_CS signal to the external memory or peripheral is asserted. The CSPMCR register controls muxing of FB_TA with other signals. If auto-acknowledge is not used and CSPMCR does not allow FB_TA control, FlexBus may hang.
FB_CLK	O	FlexBus Clock Output

32.3 Memory Map/Register Definition

The following tables describe the registers and bit meanings for configuring chip-select operation.

The actual number of chip selects available depends upon the device and its pin configuration. If the device does not support certain chip select signals or the pin is not configured for a chip-select function, then that corresponding set of chip-select registers has no effect on an external pin.

Note

You must set CSMR0[V] before the chip select registers take effect.

A bus error occurs when writing to reserved register locations.

FB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_E000	Chip Select Address Register (FB_CSAR0)	32	R/W	0000_0000h	32.3.1/1338
4001_E004	Chip Select Mask Register (FB_CSMR0)	32	R/W	0000_0000h	32.3.2/1339
4001_E008	Chip Select Control Register (FB_CSCR0)	32	R/W	0000_0000h	32.3.3/1340
4001_E00C	Chip Select Address Register (FB_CSAR1)	32	R/W	0000_0000h	32.3.1/1338
4001_E010	Chip Select Mask Register (FB_CSMR1)	32	R/W	0000_0000h	32.3.2/1339
4001_E014	Chip Select Control Register (FB_CSCR1)	32	R/W	0000_0000h	32.3.3/1340
4001_E018	Chip Select Address Register (FB_CSAR2)	32	R/W	0000_0000h	32.3.1/1338
4001_E01C	Chip Select Mask Register (FB_CSMR2)	32	R/W	0000_0000h	32.3.2/1339
4001_E020	Chip Select Control Register (FB_CSCR2)	32	R/W	0000_0000h	32.3.3/1340
4001_E024	Chip Select Address Register (FB_CSAR3)	32	R/W	0000_0000h	32.3.1/1338
4001_E028	Chip Select Mask Register (FB_CSMR3)	32	R/W	0000_0000h	32.3.2/1339
4001_E02C	Chip Select Control Register (FB_CSCR3)	32	R/W	0000_0000h	32.3.3/1340
4001_E030	Chip Select Address Register (FB_CSAR4)	32	R/W	0000_0000h	32.3.1/1338
4001_E034	Chip Select Mask Register (FB_CSMR4)	32	R/W	0000_0000h	32.3.2/1339
4001_E038	Chip Select Control Register (FB_CSCR4)	32	R/W	0000_0000h	32.3.3/1340
4001_E060	Chip Select port Multiplexing Control Register (FB_CSPMCR)	32	R/W	0000_0000h	32.3.4/1343

32.3.1 Chip Select Address Register (FB_CSAR*n*)

Specifies the associated chip-select's base address.

Address: 4001_E000h base + 0h offset + (12d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BA																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FB_CSAR*n* field descriptions

Field	Description
31–16 BA	Base Address Defines the base address for memory dedicated to the associated chip-select. BA is compared to bits 31–16 on the internal address bus to determine if the associated chip-select's memory is being accessed. **NOTE:** Because the FlexBus module is one of the slaves connected to the crossbar switch, it is only accessible within a certain memory range. See the chip memory map for the applicable FlexBus "expansion" address range for which the chip-selects can be active. Set the CSARn and CSMRn registers appropriately before accessing this region.

Table continues on the next page...

FB_CSAR*n* field descriptions (continued)

Field	Description
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

32.3.2 Chip Select Mask Register (FB_CSMR*n*)

Specifies the address mask and allowable access types for the associated chip-select.

Address: 4001_E000h base + 4h offset + (12d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BAM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							WP	0							V
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FB_CSMR*n* field descriptions

Field	Description
31–16 BAM	Base Address Mask Defines the associated chip-select's block size by masking address bits. 0 The corresponding address bit in CSAR is used in the chip-select decode. 1 The corresponding address bit in CSAR is a don't care in the chip-select decode.
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 WP	Write Protect Controls write accesses to the address range in the corresponding CSAR. 0 Write accesses are allowed. 1 Write accesses are not allowed. Attempting to write to the range of addresses for which the WP bit is set results in a bus error termination of the internal cycle and no external cycle.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 V	Valid Specifies whether the corresponding CSAR, CSMR, and CSCR contents are valid. Programmed chip-selects do not assert until the V bit is 1b (except for FB_CS0, which acts as the global chip-select). **NOTE:** At reset, $\overline{\text{FB\_CS0}}$ will fire for any access to the FlexBus memory region. CSMR0[V] must be set as part of the chip select initialization sequence to allow other chip selects to function as programmed.

Table continues on the next page...

FB_CSMRn field descriptions (continued)

Field	Description
	0 Chip-select is invalid. 1 Chip-select is valid.

32.3.3 Chip Select Control Register (FB_CSCRn)

Controls the auto-acknowledge, address setup and hold times, port size, burst capability, and number of wait states for the associated chip select.

NOTE

To support the global chip-select ($\overline{FB\_CS0}$), the CSCR0 reset values differ from the other CSCRs. The reset value of CSCR0 is as follows:

- Bits 31–24 are 0b
- Bit 23–3 are chip-dependent
- Bits 3–0 are 0b

See the chip configuration details for your particular chip for information on the exact CSCR0 reset value.

Address: 4001_E000h base + 8h offset + (12d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SWS						0		SWSEN	EXTS	ASET		RDAH		WRAH	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WS						BLS	AA	PS		BEM	BSTR	BSTW	0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FB_CSCRn field descriptions

Field	Description
31–26 SWS	Secondary Wait States Used only when the SWSEN bit is 1b. Specifies the number of wait states inserted before an internal transfer acknowledge is generated for a burst transfer (except for the first termination, which is controlled by WS).

Table continues on the next page...

FB_CSCR*n* field descriptions (continued)

Field	Description
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 SWSEN	Secondary Wait State Enable 0 Disabled. A number of wait states (specified by WS) are inserted before an internal transfer acknowledge is generated for all transfers. 1 Enabled. A number of wait states (specified by SWS) are inserted before an internal transfer acknowledge is generated for burst transfer secondary terminations.
22 EXTS	Extended Transfer Start/Extended Address Latch Enable Controls how long $\overline{\text{FB\_TS}}$ /FB_ALE is asserted. 0 Disabled. $\overline{\text{FB\_TS}}$ /FB_ALE asserts for one bus clock cycle. 1 Enabled. $\overline{\text{FB\_TS}}$ /FB_ALE remains asserted until the first positive clock edge after $\overline{\text{FB\_CSn}}$ asserts.
21–20 ASET	Address Setup Controls when the chip-select is asserted with respect to assertion of a valid address and attributes. 00 Assert $\overline{\text{FB\_CSn}}$ on the first rising clock edge after the address is asserted (default for all but $\overline{\text{FB\_CS0}}$). 01 Assert $\overline{\text{FB\_CSn}}$ on the second rising clock edge after the address is asserted. 10 Assert $\overline{\text{FB\_CSn}}$ on the third rising clock edge after the address is asserted. 11 Assert $\overline{\text{FB\_CSn}}$ on the fourth rising clock edge after the address is asserted (default for $\overline{\text{FB\_CS0}}$).
19–18 RDAH	Read Address Hold or Deselect Controls the address and attribute hold time after the termination during a read cycle that hits in the associated chip-select's address space. **NOTE:** • The hold time applies only at the end of a transfer. Therefore, during a burst transfer or a transfer to a port size smaller than the transfer size, the hold time is only added after the last bus cycle. • The number of cycles the address and attributes are held after $\overline{\text{FB\_CSn}}$ deassertion depends on the value of the AA bit. 00 When AA is 0b, 1 cycle. When AA is 1b, 0 cycles. 01 When AA is 0b, 2 cycles. When AA is 1b, 1 cycle. 10 When AA is 0b, 3 cycles. When AA is 1b, 2 cycles. 11 When AA is 0b, 4 cycles. When AA is 1b, 3 cycles.
17–16 WRAH	Write Address Hold or Deselect Controls the address, data, and attribute hold time after the termination of a write cycle that hits in the associated chip-select's address space. **NOTE:** The hold time applies only at the end of a transfer. Therefore, during a burst transfer or a transfer to a port size smaller than the transfer size, the hold time is only added after the last bus cycle. 00 1 cycle (default for all but $\overline{\text{FB\_CS0}}$) 01 2 cycles 10 3 cycles 11 4 cycles (default for $\overline{\text{FB\_CS0}}$)
15–10 WS	Wait States

Table continues on the next page...

FB_CSCR*n* field descriptions (continued)

Field	Description
	Specifies the number of wait states inserted after FlexBus asserts the associated chip-select and before an internal transfer acknowledge is generated (WS = 00h inserts 0 wait states, ..., WS = 3Fh inserts 63 wait states).
9 BLS	Byte-Lane Shift Specifies if data on FB_AD appears left-aligned or right-aligned during the data phase of a FlexBus access. 0 Not shifted. Data is left-aligned on FB_AD. 1 Shifted. Data is right-aligned on FB_AD.
8 AA	Auto-Acknowledge Enable Asserts the internal transfer acknowledge for accesses specified by the chip-select address. **NOTE:** If AA is 1b for a corresponding FB_CSn and the external system asserts an external $\overline{\text{FB\_TA}}$ before the wait-state countdown asserts the internal FB_TA, the cycle is terminated. Burst cycles increment the address bus between each internal termination. **NOTE:** This field must be 1b if CSPMCR disables FB_TA. 0 Disabled. No internal transfer acknowledge is asserted and the cycle is terminated externally. 1 Enabled. Internal transfer acknowledge is asserted as specified by WS.
7–6 PS	Port Size Specifies the data port width of the associated chip-select, and determines where data is driven during write cycles and where data is sampled during read cycles. 00 32-bit port size. Valid data is sampled and driven on FB_D[31:0]. 01 8-bit port size. Valid data is sampled and driven on FB_D[31:24] when BLS is 0b, or FB_D[7:0] when BLS is 1b. 1X 16-bit port size. Valid data is sampled and driven on FB_D[31:16] when BLS is 0b, or FB_D[15:0] when BLS is 1b.
5 BEM	Byte-Enable Mode Specifies whether the corresponding $\overline{\text{FB\_BE}}$ is asserted for read accesses. Certain memories have byte enables that must be asserted during reads and writes. Write 1b to the BEM bit in the relevant CSCR to provide the appropriate mode of byte enable support for these SRAMs. 0 $\overline{\text{FB\_BE}}$ is asserted for data write only. 1 $\overline{\text{FB\_BE}}$ is asserted for data read and write accesses.
4 BSTR	Burst-Read Enable Specifies whether burst reads are enabled for memory associated with each chip select. 0 Disabled. Data exceeding the specified port size is broken into individual, port-sized, non-burst reads. For example, a 32-bit read from an 8-bit port is broken into four 8-bit reads. 1 Enabled. Enables data burst reads larger than the specified port size, including 32-bit reads from 8- and 16-bit ports, 16-bit reads from 8-bit ports, and line reads from 8-, 16-, and 32-bit ports.
3 BSTW	Burst-Write Enable Specifies whether burst writes are enabled for memory associated with each chip select.

Table continues on the next page...

FB_CSCR*n* field descriptions (continued)

Field	Description
	0 Disabled. Data exceeding the specified port size is broken into individual, port-sized, non-burst writes. For example, a 32-bit write to an 8-bit port takes four byte writes. 1 Enabled. Enables burst write of data larger than the specified port size, including 32-bit writes to 8- and 16-bit ports, 16-bit writes to 8-bit ports, and line writes to 8-, 16-, and 32-bit ports.
2–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

32.3.4 Chip Select port Multiplexing Control Register (FB_CSPMCR)

Controls the multiplexing of the FlexBus signals.

NOTE

A bus error occurs when you do any of the following:

- Write to a reserved address
- Write to a reserved field in this register, or
- Access this register using a size other than 32 bits.

Address: 4001_E000h base + 60h offset = 4001_E060h

Bit	31–28	27–24	23–20	19–16	15–12	11–0
R	GROUP1	GROUP2	GROUP3	GROUP4	GROUP5	0
W						
Reset	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0

FB_CSPMCR field descriptions

Field	Description
31–28 GROUP1	FlexBus Signal Group 1 Multiplex control Controls the multiplexing of the FB_ALE, $\overline{\text{FB\_CS1}}$, and $\overline{\text{FB\_TS}}$ signals. 0000 FB_ALE 0001 $\overline{\text{FB\_CS1}}$ 0010 $\overline{\text{FB\_TS}}$ Any other value Reserved
27–24 GROUP2	FlexBus Signal Group 2 Multiplex control Controls the multiplexing of the $\overline{\text{FB\_CS4}}$, FB_TSIZ0, and $\overline{\text{FB\_BE\_31\_24}}$ signals. 0000 $\overline{\text{FB\_CS4}}$ 0001 FB_TSIZ0 0010 $\overline{\text{FB\_BE\_31\_24}}$ Any other value Reserved
23–20 GROUP3	FlexBus Signal Group 3 Multiplex control Controls the multiplexing of the $\overline{\text{FB\_CS5}}$, FB_TSIZ1, and $\overline{\text{FB\_BE\_23\_16}}$ signals.

Table continues on the next page...

FB_CSPMCR field descriptions (continued)

Field	Description
	0000 FB_CS5 0001 FB_TSIZ1 0010 FB_BE_23_16 Any other value Reserved
19–16 GROUP4	FlexBus Signal Group 4 Multiplex control Controls the multiplexing of the FB_TBST , FB_CS2 , and FB_BE_15_8 signals. 0000 FB_TBST 0001 FB_CS2 0010 FB_BE_15_8 Any other value Reserved
15–12 GROUP5	FlexBus Signal Group 5 Multiplex control Controls the multiplexing of the FB_TA , FB_CS3 , and FB_BE_7_0 signals. **NOTE:** When GROUP5 is not 0000b, you must write 1b to the CSCR[AA] bit. Otherwise, the bus hangs during a transfer. 0000 FB_TA 0001 FB_CS3 . You must also write 1b to CSCR[AA]. 0010 FB_BE_7_0 . You must also write 1b to CSCR[AA]. Any other value Reserved
11–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

32.4 Functional description

32.4.1 Modes of operation

FlexBus supports the following modes of operation:

- Multiplexed 32-bit address and 32-bit data
- Multiplexed 32-bit address and 16-bit data (non-multiplexed 16-bit address and 16-bit data)
- Multiplexed 32-bit address and 8-bit data (non-multiplexed 24-bit address and 8-bit data)
- Non-multiplexed 32-bit address and 32-bit data busses

32.4.2 Address comparison

When a bus cycle is routed to FlexBus, FlexBus compares the transfer address to the base address (see CSAR[BA]) and base address mask (see CSMR[BAM]). This table describes how FlexBus decides to assert a chip-select and complete the bus cycle based on the address comparison.

When the transfer address	Then FlexBus
Matches one address register configuration	Asserts the appropriate chip-select, generating a FlexBus bus cycle as defined in the appropriate CSCR. If CSMR[WP] is set and a write access is performed, FlexBus terminates the internal bus cycle with a bus error, does not assert a chip-select, and does not perform an external bus cycle.
Does not match an address register configuration	Terminates the transfer with a bus error response, does not assert a chip-select, and does not perform a FlexBus cycle.
Matches more than one address register configuration	Terminates the transfer with a bus error response, does not assert a chip-select, and does not perform a FlexBus cycle.

32.4.3 Address driven on address bus

FlexBus always drives a 32-bit address on the FB_AD bus regardless of the external memory's or peripheral's address size.

32.4.4 Connecting address/data lines

The external device must connect its address and data lines as follows:

- Address lines
  - FB_AD from FB_AD0 upward
- Data lines
  - If CSCR[BLS] = 0, FB_AD from FB_AD31 downward
  - If CSCR[BLS] = 1, FB_AD from FB_AD0 upward

32.4.5 Bit ordering

No bit ordering is required when connecting address and data lines to the FB_AD bus. For example, a full 16-bit address/16-bit data device connects its addr15–addr0 to FB_AD16–FB_AD1 and data15–data0 to FB_AD31–FB_AD16. See Data-byte alignment and physical connections for a graphical connection.

32.4.6 Data transfer signals

Data transfers between FlexBus and the external memory or peripheral involve these signals:

- Address/data bus (FB_AD31–FB_AD0)
- Control signals ($\overline{\text{FB\_TS}}$/FB_ALE, $\overline{\text{FB\_TA}}$, $\overline{\text{FB\_CS}n}$, $\overline{\text{FB\_OE}}$, FB_R/$\overline{\text{W}}$, $\overline{\text{FB\_BE}n}$)
- Attribute signals ($\overline{\text{FB\_TBST}}$, FB_TSIZ1–FB_TSIZ0)

32.4.7 Signal transitions

These signals change on the rising edge of the FlexBus clock (FB_CLK):

- Address
- Write data
- $\overline{\text{FB\_TS}}$/FB_ALE
- $\overline{\text{FB\_CSn}}$
- All attribute signals

FlexBus latches the read data on the rising edge of the clock.

32.4.8 Data-byte alignment and physical connections

The device aligns data transfers in FlexBus byte lanes with the number of lanes depending on the data port width.

The following figure shows the byte lanes that external memory or peripheral connects to and the sequential transfers of a 32-bit transfer for the supported port sizes when byte lane shift is disabled. For example, an 8-bit memory connects to the single lane FB_AD31–FB_AD24 ($\overline{\text{FB\_BE\_31\_24}}$). A 32-bit transfer through this 8-bit port takes four transfers, starting with the LSB to the MSB. A 32-bit transfer through a 32-bit port requires one transfer on each four-byte lane.

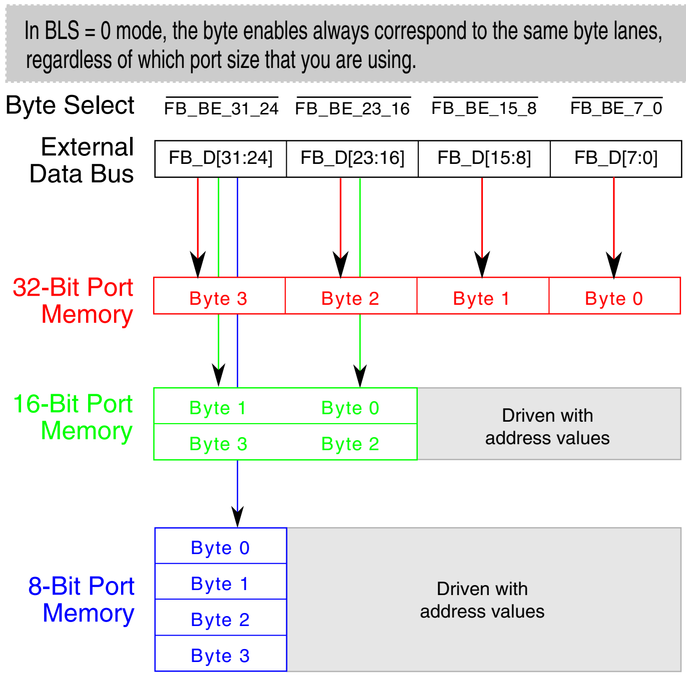


Figure 32-20. Connections for external memory port sizes (CSCRn[BLS] = 0)

The following figure shows the byte lanes that external memory or peripheral connects to and the sequential transfers of a 32-bit transfer for the supported port sizes when byte lane shift is enabled.

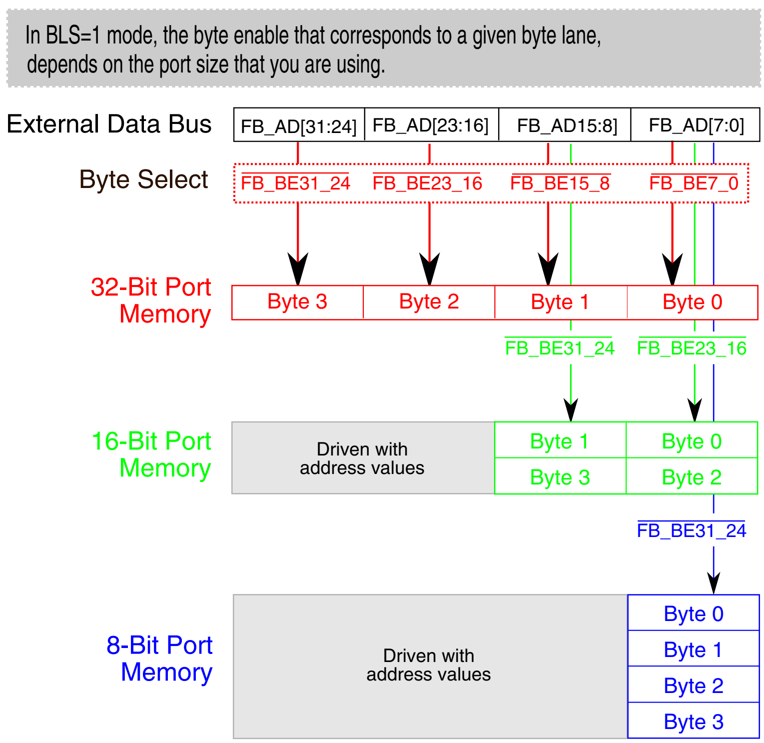


Figure 32-21. Connections for external memory port sizes (CSCRn[BLS] = 1)

32.4.9 Address/data bus multiplexing

FlexBus supports a single 32-bit wide multiplexed address and data bus (FB_AD31–FB_AD0). FlexBus always drives the full 32-bit address on the first clock of a bus cycle. During the data phase, the FB_AD31– FB_AD0 lines used for data are determined by the programmed port size and BLS setting for the corresponding chip-select. FlexBus continues to drive the address on any FB_AD31– FB_AD0 lines not used for data.

32.4.9.1 FlexBus multiplexed operating modes for CSCRn[BLS]=0

This table shows the supported combinations of address and data bus widths when CSCRn[BLS] is 0b.

<table>
<tr><th colspan="2" rowspan="2">Port size and phase</th><th colspan="4">FB_AD</th></tr>
<tr><th>31–24</th><th>23–16</th><th>15–8</th><th>7–0</th></tr>
<tr><td rowspan="2">32-bit</td><td>Address phase</td><td colspan="4">Address</td></tr>
<tr><td>Data phase</td><td colspan="4">Data</td></tr>
<tr><td rowspan="2">16-bit</td><td>Address phase</td><td colspan="4">Address</td></tr>
<tr><td>Data phase</td><td colspan="2">Data</td><td colspan="2">Address</td></tr>
<tr><td rowspan="2">8-bit</td><td>Address phase</td><td colspan="4">Address</td></tr>
<tr><td>Data phase</td><td>Data</td><td colspan="3">Address</td></tr>
</table>

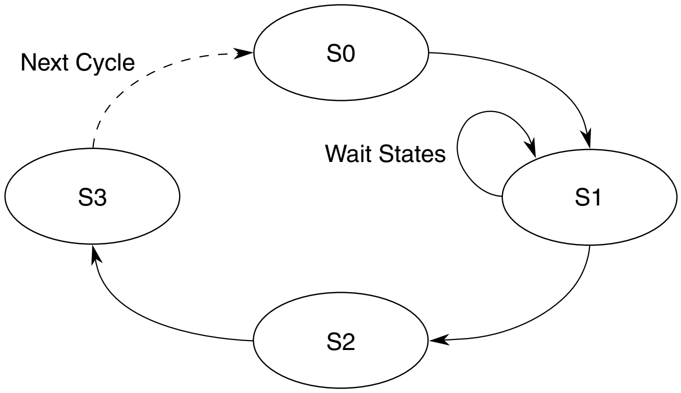
32.4.9.2 FlexBus multiplexed operating modes for CSCRn[BLS]=1

This table shows the supported combinations of address and data bus widths when CSCRn[BLS] is 1b.

<table>
<tr><th colspan="2" rowspan="2">Port size and phase</th><th colspan="4">FB_AD</th></tr>
<tr><th>31–24</th><th>23–16</th><th>15–8</th><th>7–0</th></tr>
<tr><td rowspan="2">32-bit</td><td>Address phase</td><td colspan="4">Address</td></tr>
<tr><td>Data phase</td><td colspan="4">Data</td></tr>
<tr><td rowspan="2">16-bit</td><td>Address phase</td><td colspan="4">Address</td></tr>
<tr><td>Data phase</td><td colspan="2">Address</td><td colspan="2">Data</td></tr>
<tr><td rowspan="2">8-bit</td><td>Address phase</td><td colspan="4">Address</td></tr>
<tr><td>Data phase</td><td colspan="3">Address</td><td>Data</td></tr>
</table>

32.4.10 Data transfer states

Basic data transfers occur in four clocks or states. (See Figure 32-23 and Figure 32-25 for examples of basic data transfers.) The FlexBus state machine controls the data-transfer operation. This figure shows the state-transition diagram for basic read and write cycles.



The states are described in this table.

State	Cycle	Description
S0	All	The read or write cycle is initiated. On the rising clock edge, FlexBus: • Places a valid address on FB_ADn • Asserts FB_TS/FB_ALE • Drives FB_R/W high for a read and low for a write
S1	All	FlexBus: • Negates FB_TS/FB_ALE on the rising edge of FB_CLK • Asserts FB_CSn • Drives the data on FB_AD31– FB_ADX for writes • Tristates FB_AD31– FB_ADX for reads • Continues to drive the address on FB_AD pins that are unused for data If the external memory or perihperal asserts FB_TA, then the process moves to S2. If FB_TA is not asserted internally or externally, then S1 repeats.
	Read	The external memory or peripheral drives the data before the next rising edge of FB_CLK (the rising edge that begins S2) with FB_TA asserted.
S2	All	For internal termination, FlexBus negates FB_CSn and the transfer is complete. For external termination, the external memory or peripheral negates FB_TA, and FlexBus negates FB_CSn after the rising edge of FB_CLK at the end of S2.
	Read	FlexBus latches the data on the rising clock edge entering S2. The external memory or peripheral can stop driving the data after this edge or continue to drive the data until the end of S3 or through any additional address hold cycles.
S3	All	FlexBus invalidates the address, data, and FB_R/W on the rising edge of FB_CLK at the beginning of S3, terminating the transfer.

32.4.11 FlexBus Timing Examples

Note

The timing diagrams throughout this section use signal names that may not be included on your particular device. Ignore these extraneous signals.

Note

Throughout this section:

- FB_D[*X*] indicates a 32-, 16-, or 8-bit wide data bus
- FB_A[*Y*] indicates an address bus that can be 32, 24, or 16 bits wide.

32.4.11.1 Basic Read Bus Cycle

During a read cycle, the MCU receives data from memory or a peripheral device. The following figure shows a read cycle flowchart.

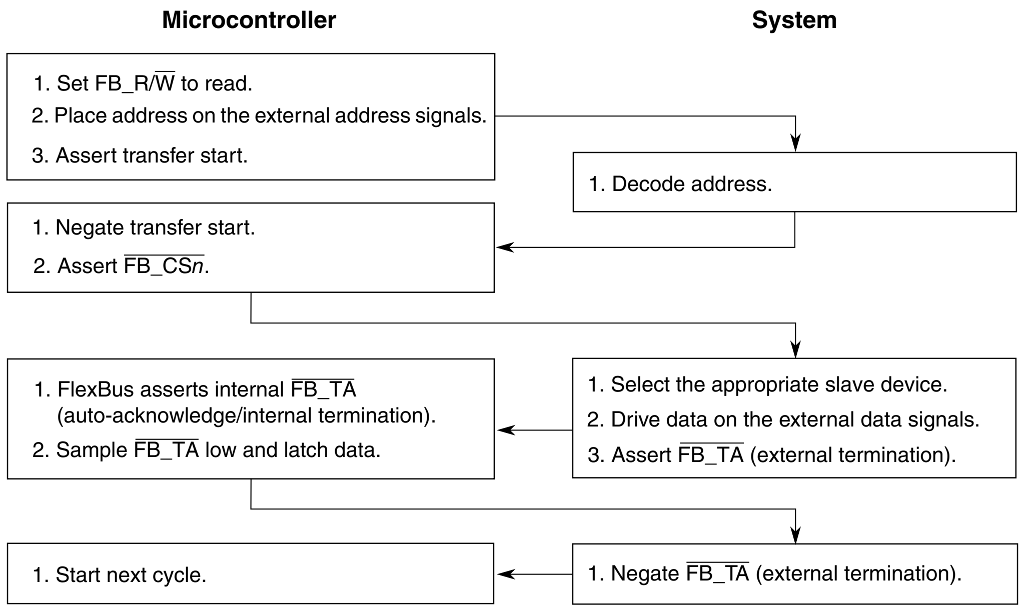


Figure 32-22. Read Cycle Flowchart

The read cycle timing diagram is shown in the following figure.

Note

FB_TA does not have to be driven by the external device for internally-terminated bus cycles.

Note

The processor drives the data lines during the first clock cycle of the transfer with the full 32-bit address. This may be ignored by standard connected devices using non-multiplexed address and data buses. However, some applications may find this feature beneficial.

The address and data busses are muxed between the FlexBus and another module. At the end of the read bus cycles the address signals are indeterminate.

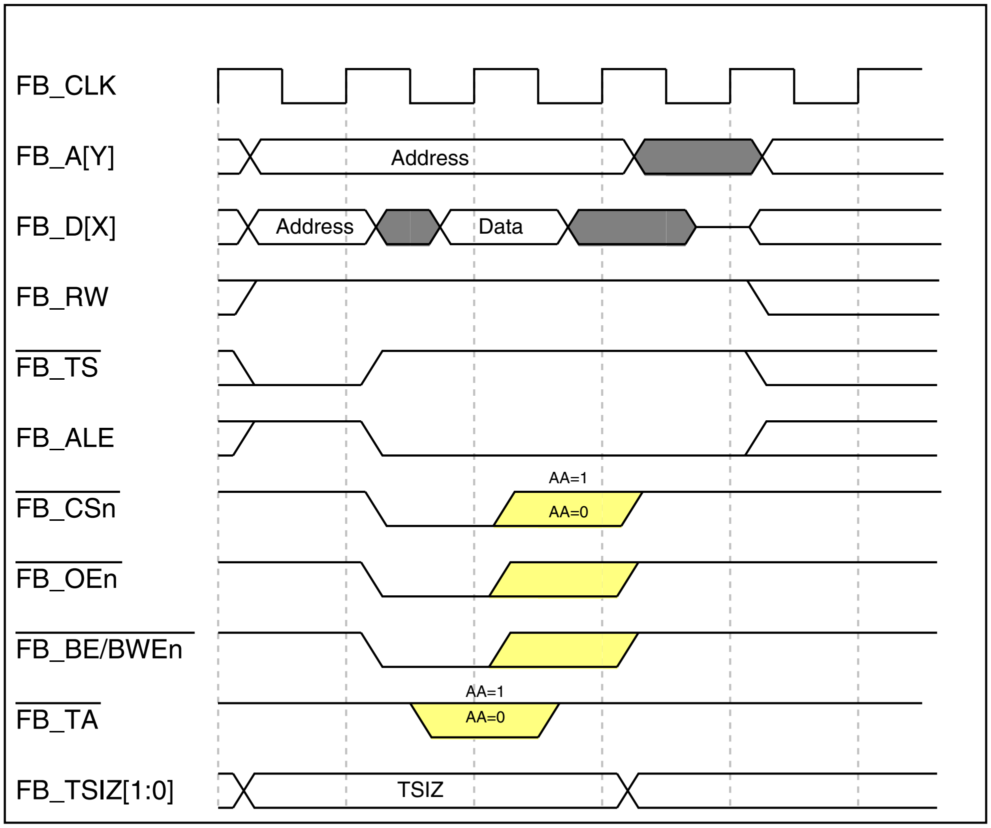


Figure 32-23. Basic Read-Bus Cycle

32.4.11.2 Basic Write Bus Cycle

During a write cycle, the device sends data to memory or to a peripheral device. The following figure shows the write cycle flowchart.

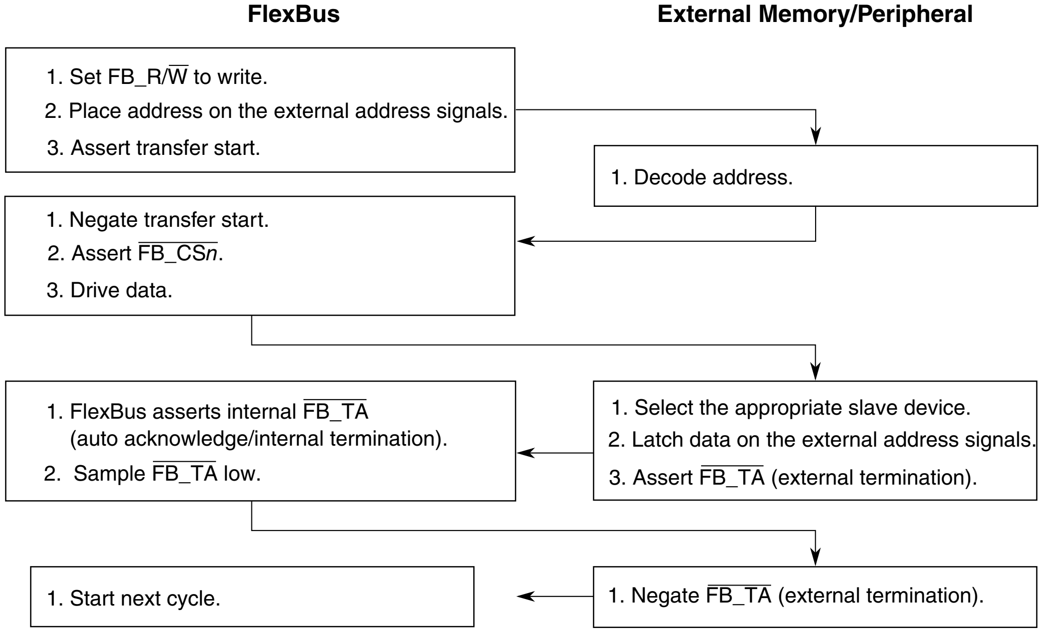


Figure 32-24. Write-Cycle Flowchart

The following figure shows the write cycle timing diagram.

Note

The address and data busses are muxed between the FlexBus and another module. At the end of the write bus cycles, the address signals are indeterminate.

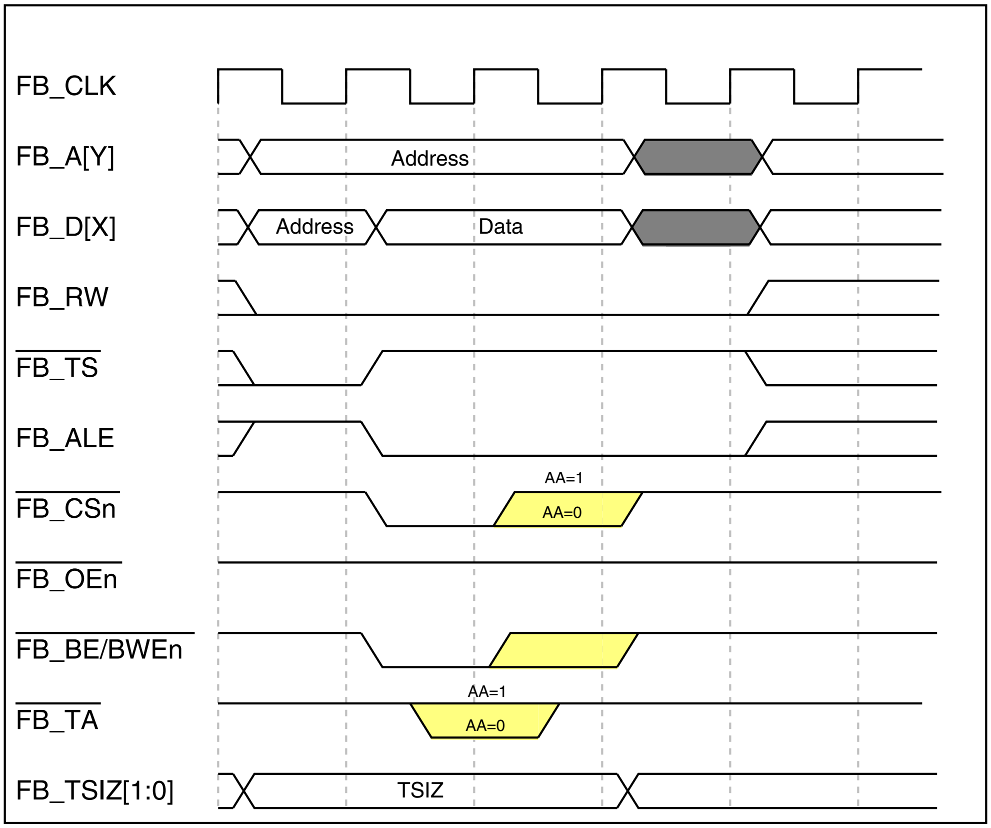


Figure 32-25. Basic Write-Bus Cycle

32.4.11.3 Bus Cycle Sizing

This section shows timing diagrams for various port size scenarios.

32.4.11.3.1 Bus Cycle Sizing—Byte Transfer, 8-bit Device, No Wait States

The following figure illustrates the basic byte read transfer to an 8-bit device with no wait states:

- The address is driven on the full FB_AD[31:8] bus in the first clock.
- The device tristates FB_AD[31:24] on the second clock and continues to drive address on FB_AD[23:0] throughout the bus cycle.
- The external device returns the read data on FB_AD[31:24] and may tristate the data line or continue driving the data one clock after FB_TA is sampled asserted.

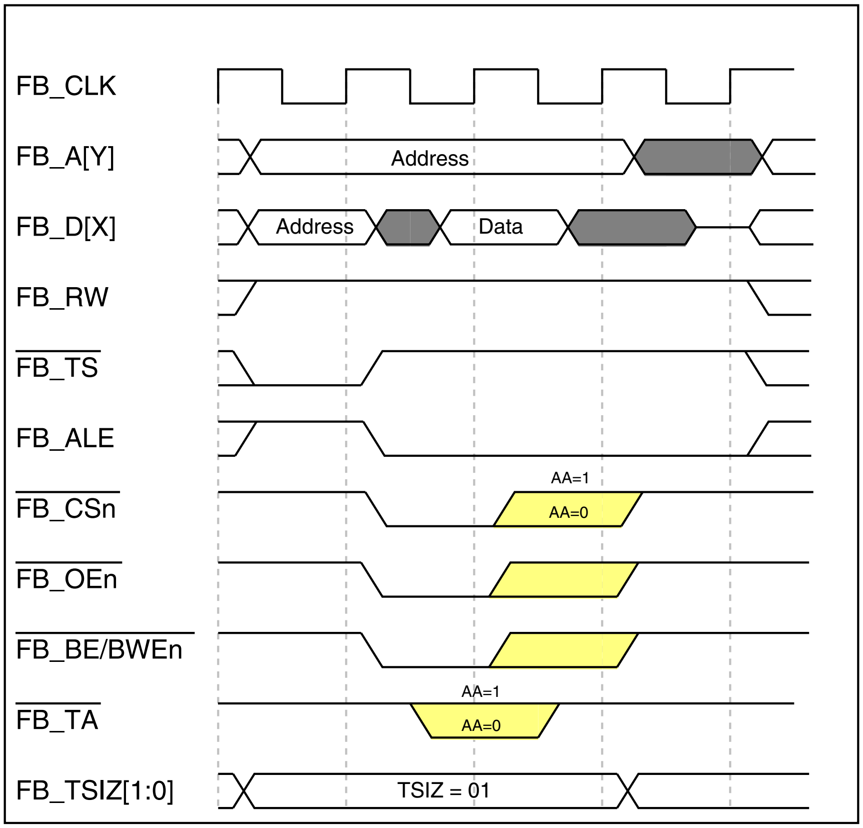


Figure 32-26. Single Byte-Read Transfer

The following figure shows the similar configuration for a write transfer. The data is driven from the second clock on FB_AD[31:24].

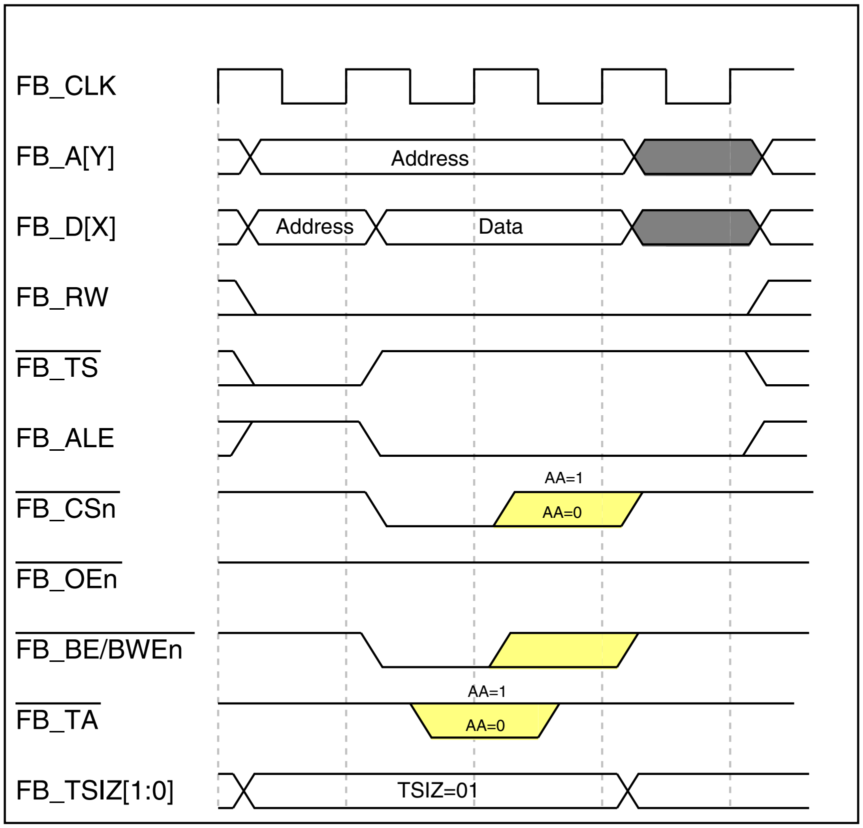


Figure 32-27. Single Byte-Write Transfer

32.4.11.3.2 Bus Cycle Sizing—Word Transfer, 16-bit Device, No Wait States

The following figure illustrates the basic word read transfer to a 16-bit device with no wait states.

- The address is driven on the full FB_AD[31:8] bus in the first clock.
- The device tristates FB_AD[31:16] on the second clock and continues to drive address on FB_AD[15:0] throughout the bus cycle.
- The external device returns the read data on FB_AD[31:16] and may tristate the data line or continue driving the data one clock after $\overline{\text{FB\_TA}}$ is sampled asserted.

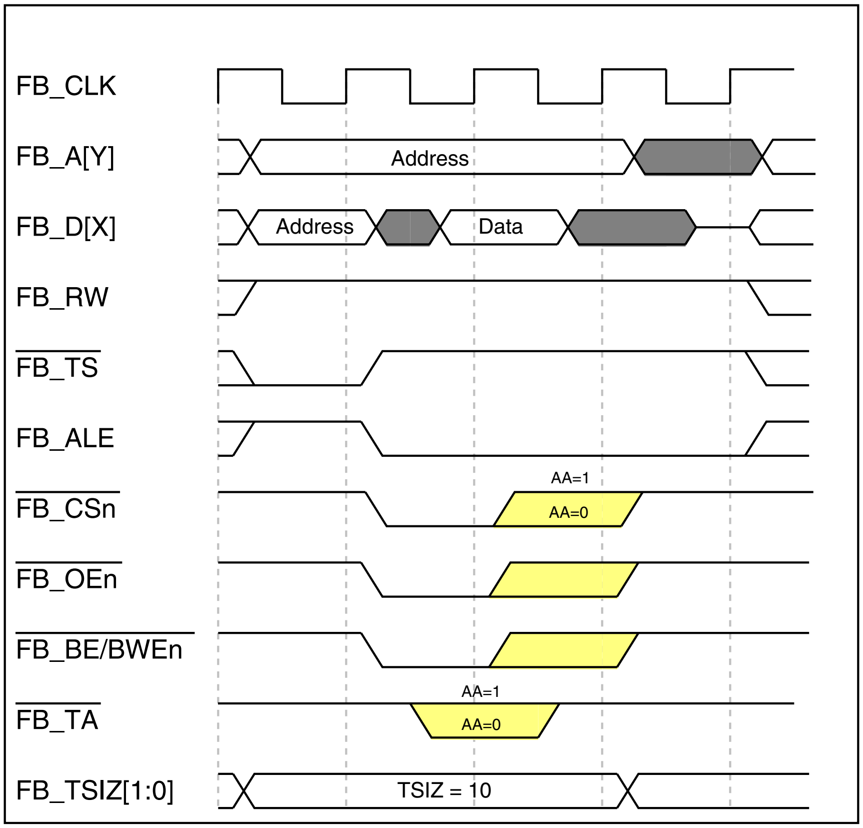


Figure 32-28. Single Word-Read Transfer

The following figure shows the similar configuration for a write transfer. The data is driven from the second clock on FB_AD[31:16].

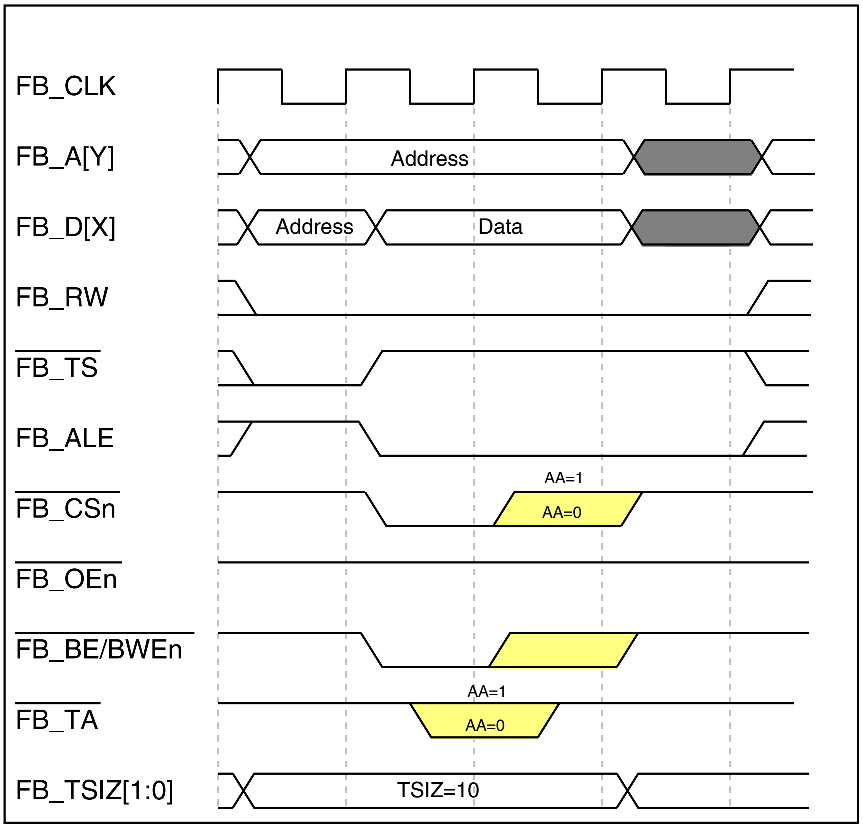


Figure 32-29. Single Word-Write Transfer

32.4.11.3.3 Bus Cycle Sizing—Longword Transfer, 32-bit Device, No Wait States

The following figure depicts a longword read from a 32-bit device.

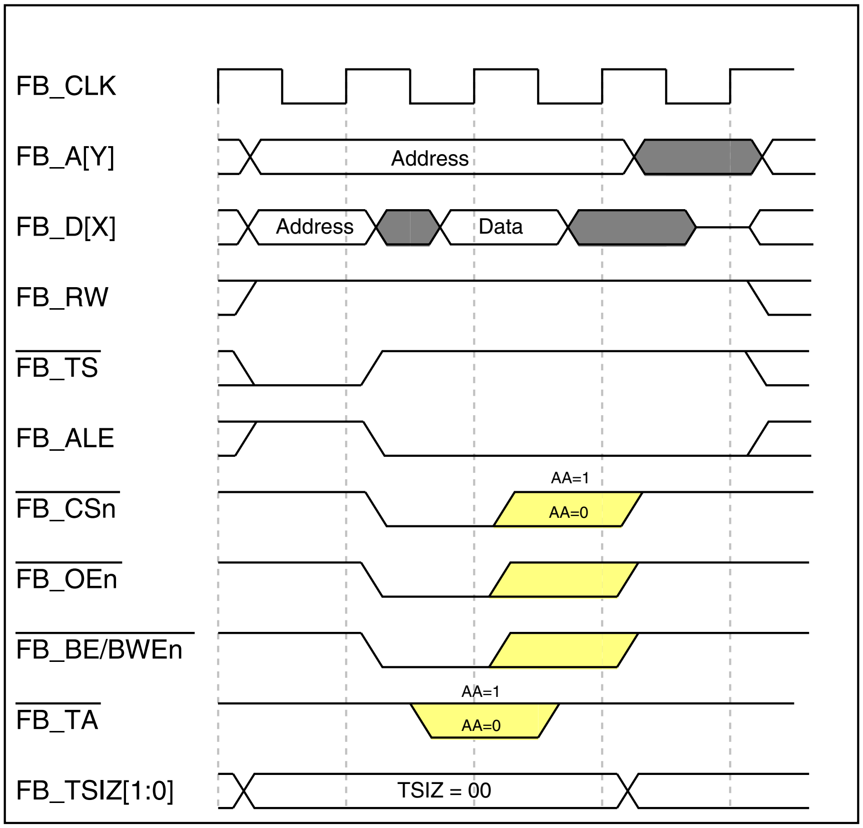


Figure 32-30. Longword-Read Transfer

The following figure illustrates the longword write to a 32-bit device.

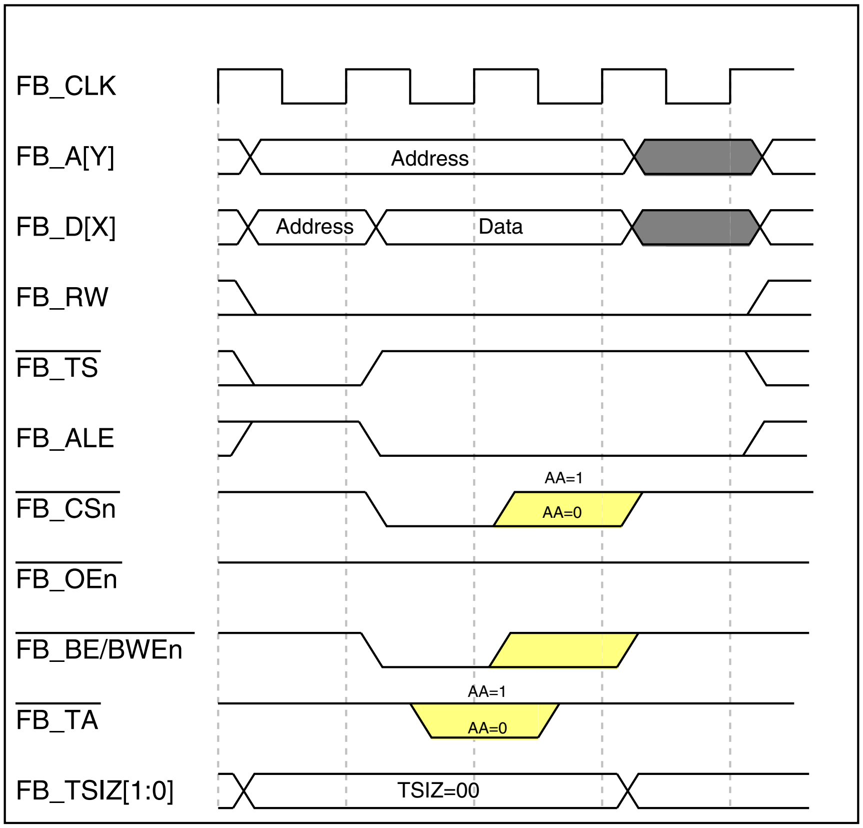


Figure 32-31. Longword-Write Transfer

32.4.11.4 Timing Variations

The FlexBus module has several features that can change the timing characteristics of a basic read- or write-bus cycle to provide additional address setup, address hold, and time for a device to provide or latch data.

32.4.11.4.1 Wait States

Wait states can be inserted before each beat of a transfer by programming the CSCR*n* registers. Wait states can give the peripheral or memory more time to return read data or sample write data.

The following figures show the basic read and write bus cycles (also shown in Figure 32-23 and Figure 32-28) with the default of no wait states respectively.

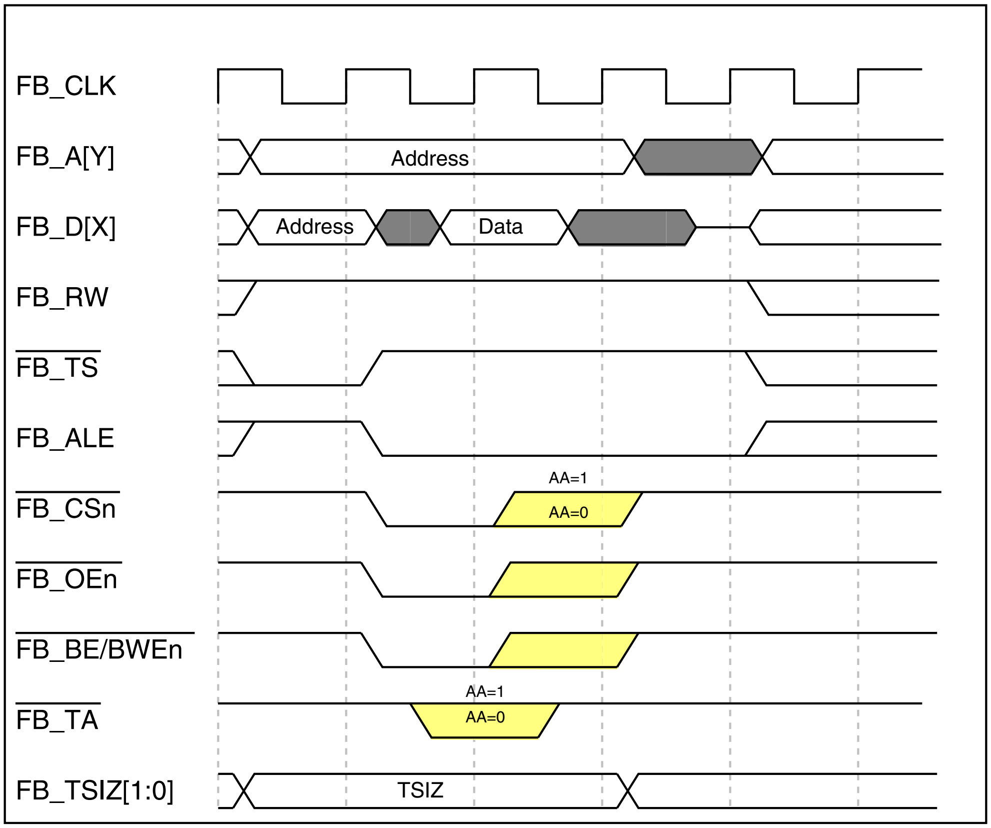


Figure 32-32. Basic Read-Bus Cycle (No Wait States)

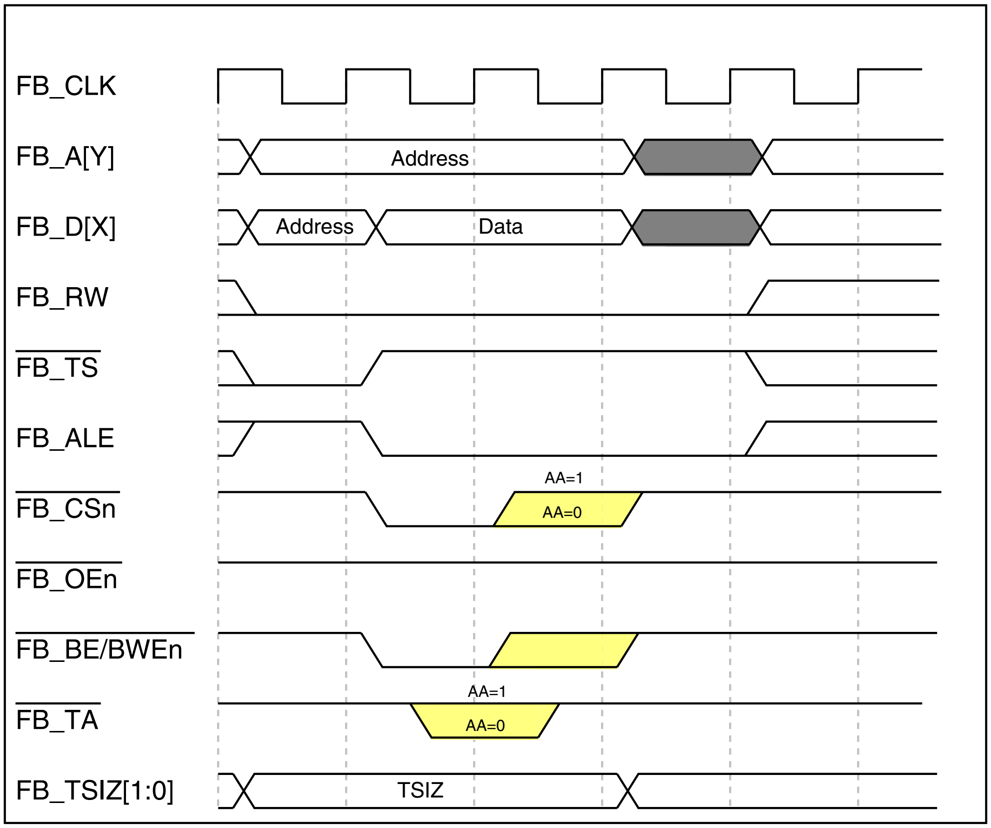


Figure 32-33. Basic Write-Bus Cycle (No Wait States)

If wait states are used, the S1 state repeats continuously until the chip-select auto-acknowledge unit asserts internal transfer acknowledge or the external $\overline{\text{FB\_TA}}$ is recognized as asserted. The following figures show a read and write cycle with one wait state respectively.

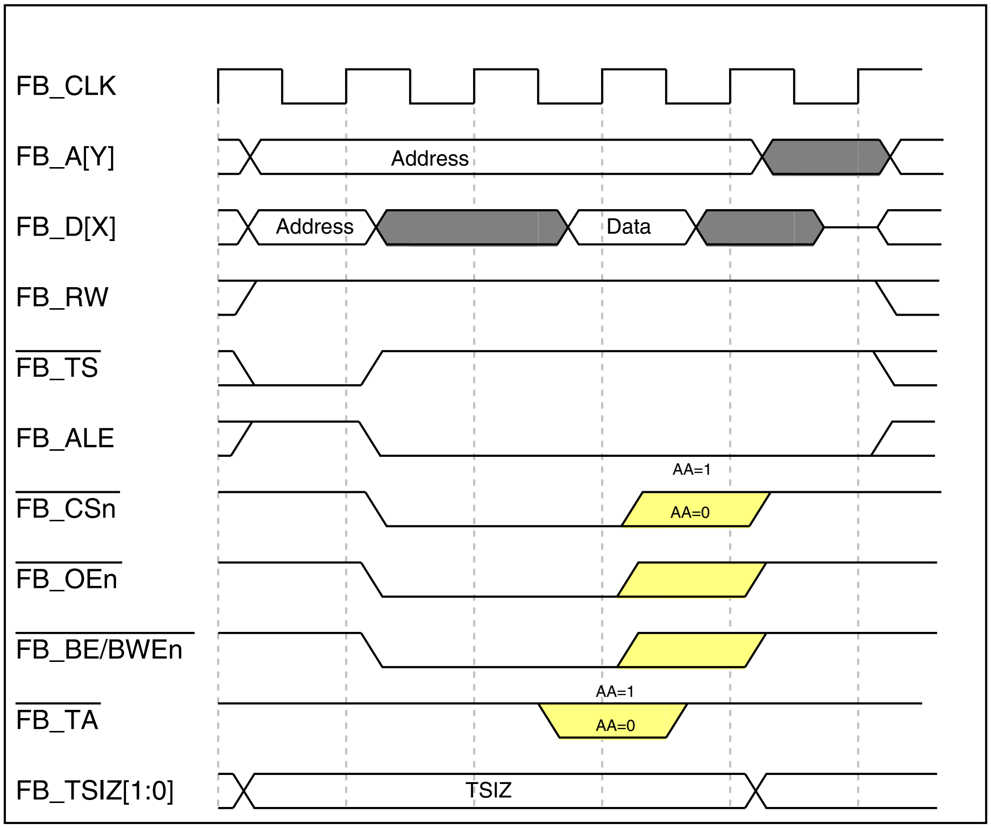


Figure 32-34. Read-Bus Cycle (One Wait State)

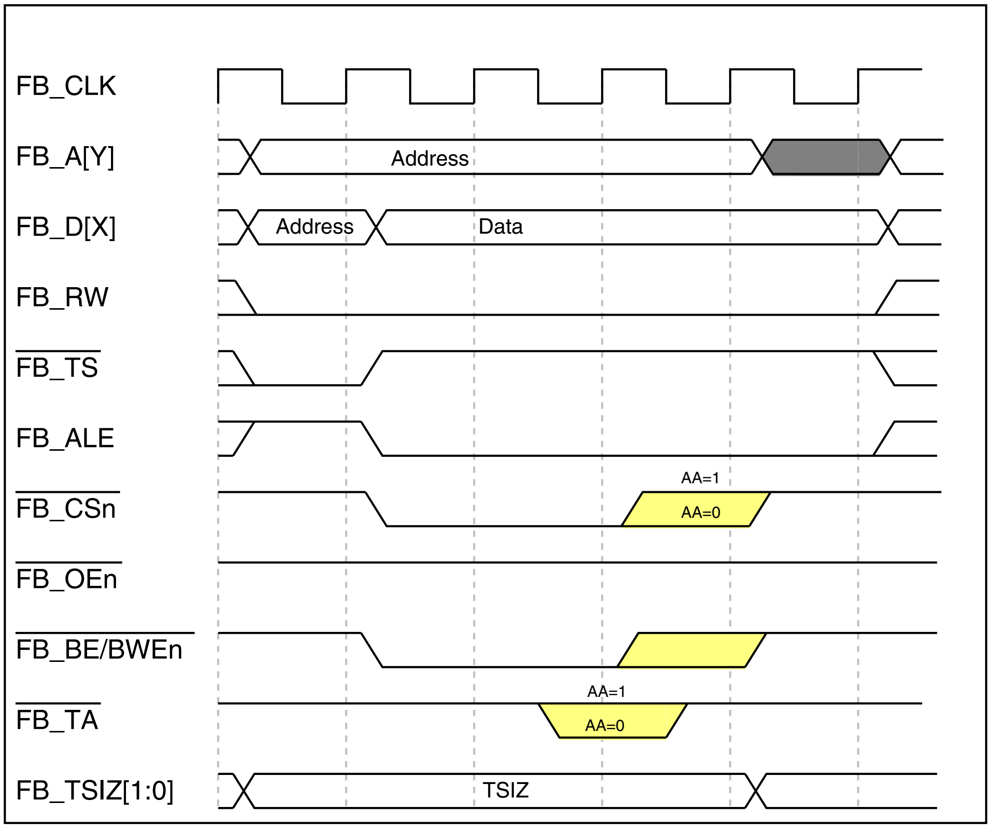


Figure 32-35. Write-Bus Cycle (One Wait State)

32.4.11.4.2 Address Setup and Hold

The timing of the assertion and negation of the chip selects, byte selects, and output enable can be programmed on a chip-select basis. Each chip-select can be programmed to assert one to four clocks after transfer start/address-latch enable (FB_TS/FB_ALE) is asserted. The following figures show read- and write-bus cycles with two clocks of address setup respectively.

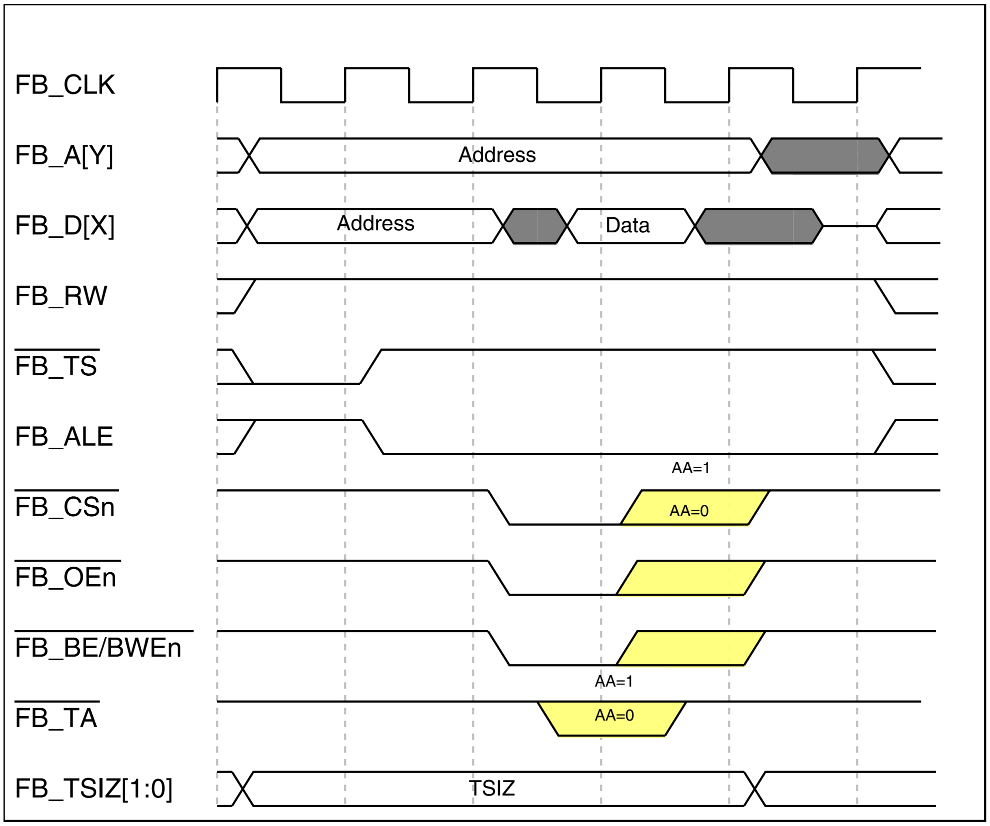


Figure 32-36. Read-Bus Cycle with Two-Clock Address Setup (No Wait States)

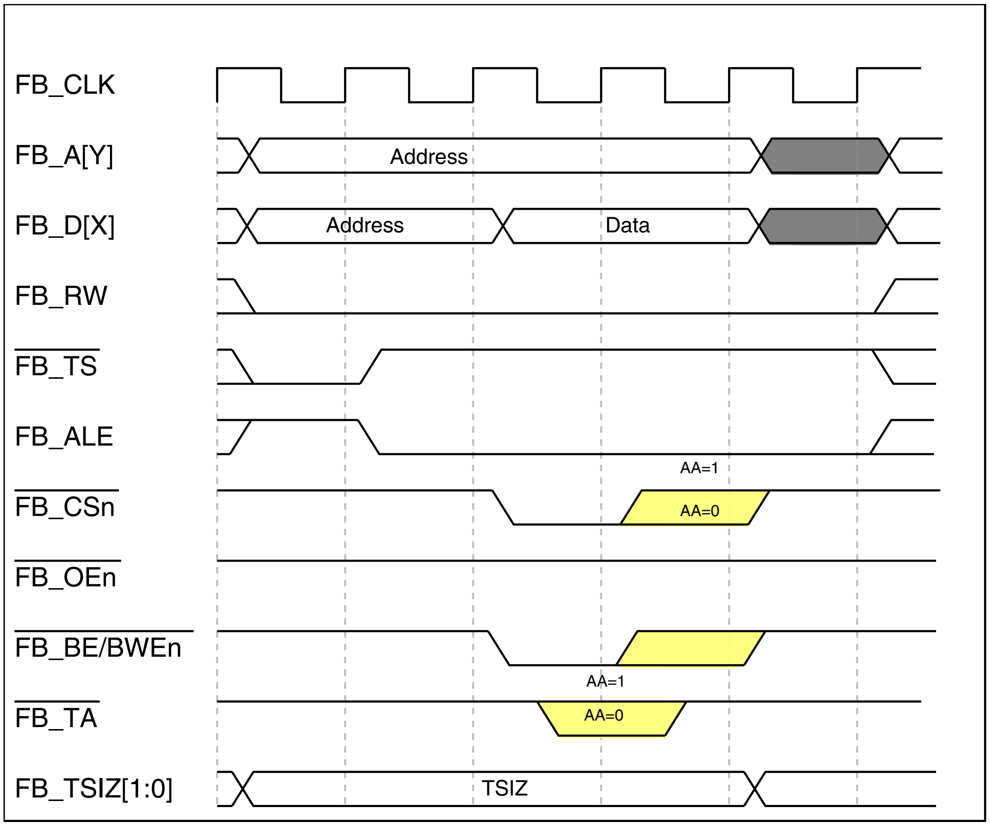


Figure 32-37. Write-Bus Cycle with Two Clock Address Setup (No Wait States)

In addition to address setup, a programmable address hold option for each chip select exists. Address and attributes can be held one to four clocks after chip-select, byte-selects, and output-enable negate. The following figures show read and write bus cycles with two clocks of address hold respectively.

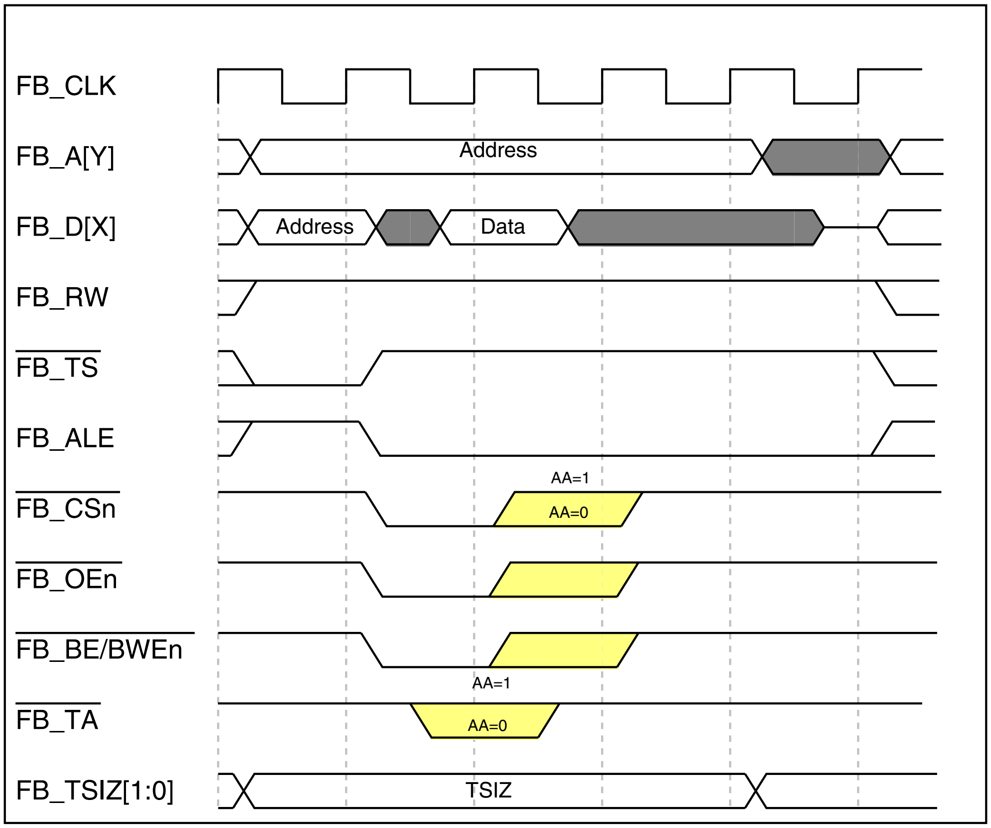


Figure 32-38. Read Cycle with Two-Clock Address Hold (No Wait States)

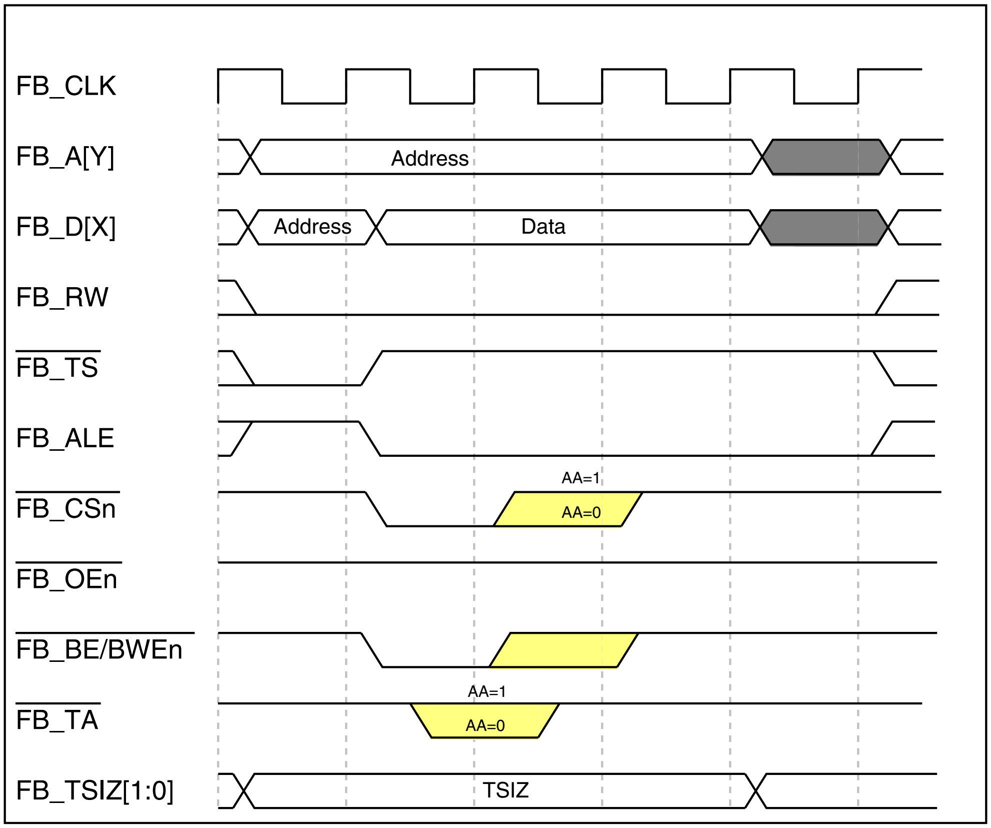


Figure 32-39. Write Cycle with Two-Clock Address Hold (No Wait States)

The following figure shows a bus cycle using address setup, wait states, and address hold.

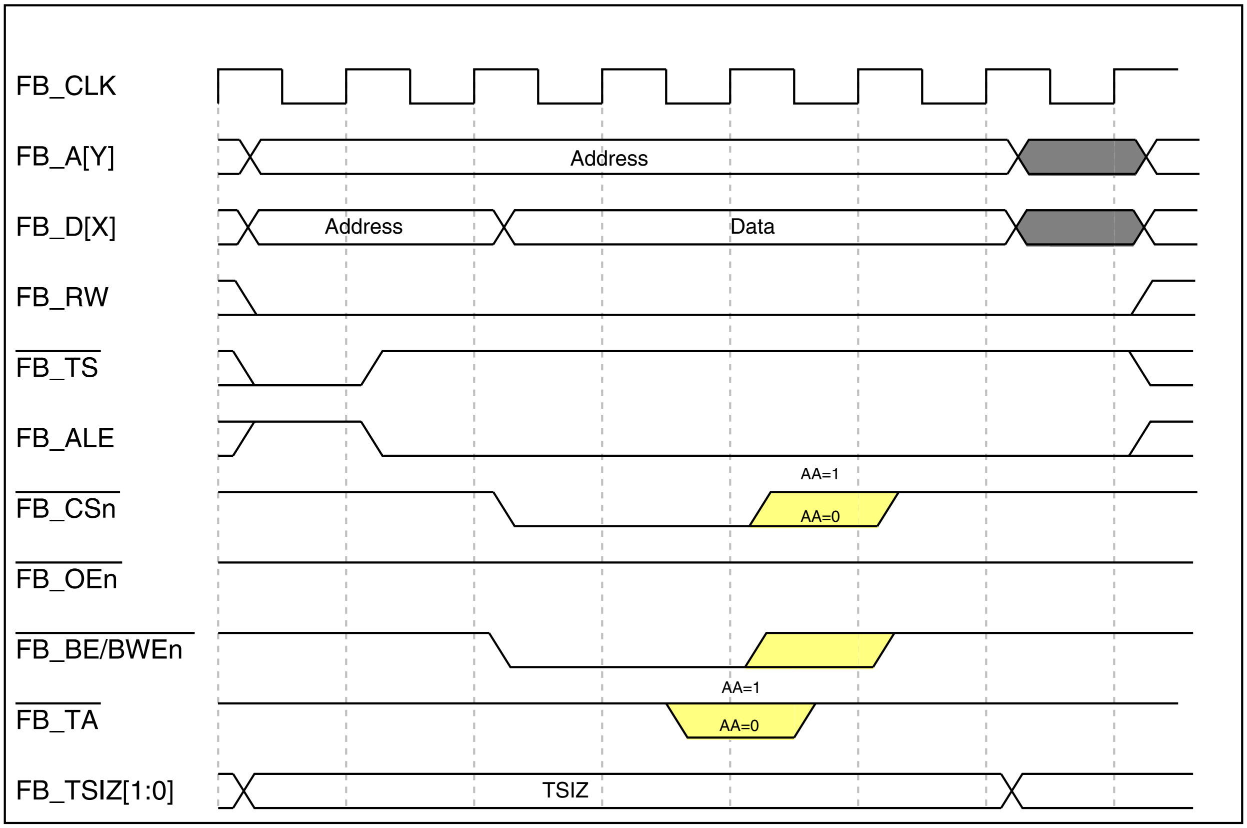


Figure 32-40. Write Cycle with Two-Clock Address Setup and Two-Clock Hold (One Wait State)

32.4.12 Burst cycles

The chip can be programmed to initiate burst cycles if its transfer size exceeds the port size of the selected destination. The initiation of a burst cycle is encoded on the transfer size pins (FB_TSIZ[1:0]). For burst transfers to smaller port sizes, FB_TSIZ[1:0] indicates the size of the entire transfer. For example, with bursting enabled, a 16-bit transfer to an 8-bit port takes two beats (two byte-sized transfers), for which FB_TSIZ[1:0] equals 10b throughout. A 32-bit transfer to an 8-bit port takes four beats (four byte-sized transfers), for which FB_TSIZ[1:0] equals 00b throughout.

32.4.12.1 Enabling and inhibiting burst

The CSCRn registers enable bursting for reads, writes, or both.

Memory spaces can be declared burst-inhibited for reads and writes by writing 0b to the appropriate CSCRn[BSTR] and CSCRn[BSTW] fields.

32.4.12.2 Transfer size and port size translation

With bursting disabled, any transfer larger than the port size breaks into multiple individual transfers (e.g. <Addr><Data><Addr+1><Data><Addr+2><Data>). With bursting enabled, any transfer larger than the port size results in a burst cycle of multiple beats (e.g. <Addr><Data><Data><Data>). The following table shows the result of such transfer translations.

Port size PS[1:0]	Transfer size FB_TSIZ[1:0]	Burst-inhibited: Number of transfers Burst enabled: Number of beats
01b (8 bit)	10b (16 bits)	2
	00b (32 bits)	4
	11b (16 bytes)	16
1Xb (16 bit)	00b (32 bits)	2
	11b (16 bytes)	8
00b (32 bit)	11b (line)	4

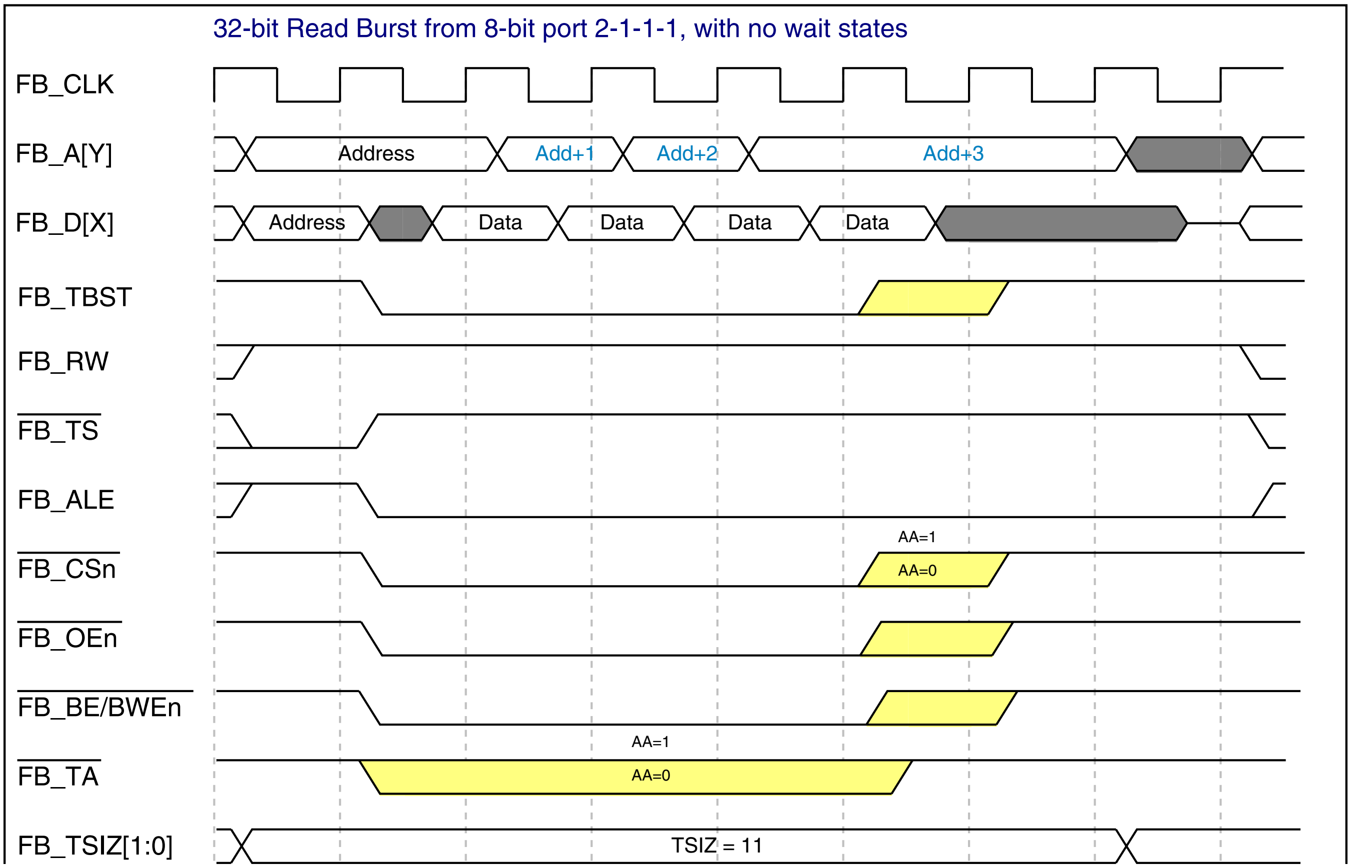
The FlexBus can support X-1-1-1 burst cycles to maximize system performance, where X is the primary number of wait states (max 63). Delaying termination of the cycle can add wait states. If internal termination is used, different wait state counters can be used for the first access and the following beats.

32.4.12.3 32-bit-Read burst from 8-Bit port 2-1-1-1 (no wait states)

The following figure shows a 32-bit read to an 8-bit external chip programmed for burst enable. The transfer results in a 4-beat burst and the data is driven on FB_AD[31:24]. The transfer size is driven at 32-bit (00b) throughout the bus cycle.

Note

> In non-multiplexed address/data mode, the address on FB_A increments only during internally-terminated burst cycles. The first address is driven throughout the entire burst for externally-terminated cycles.
>
> In multiplexed address/data mode, the address is driven on FB_AD only during the first cycle for all terminated cycles.

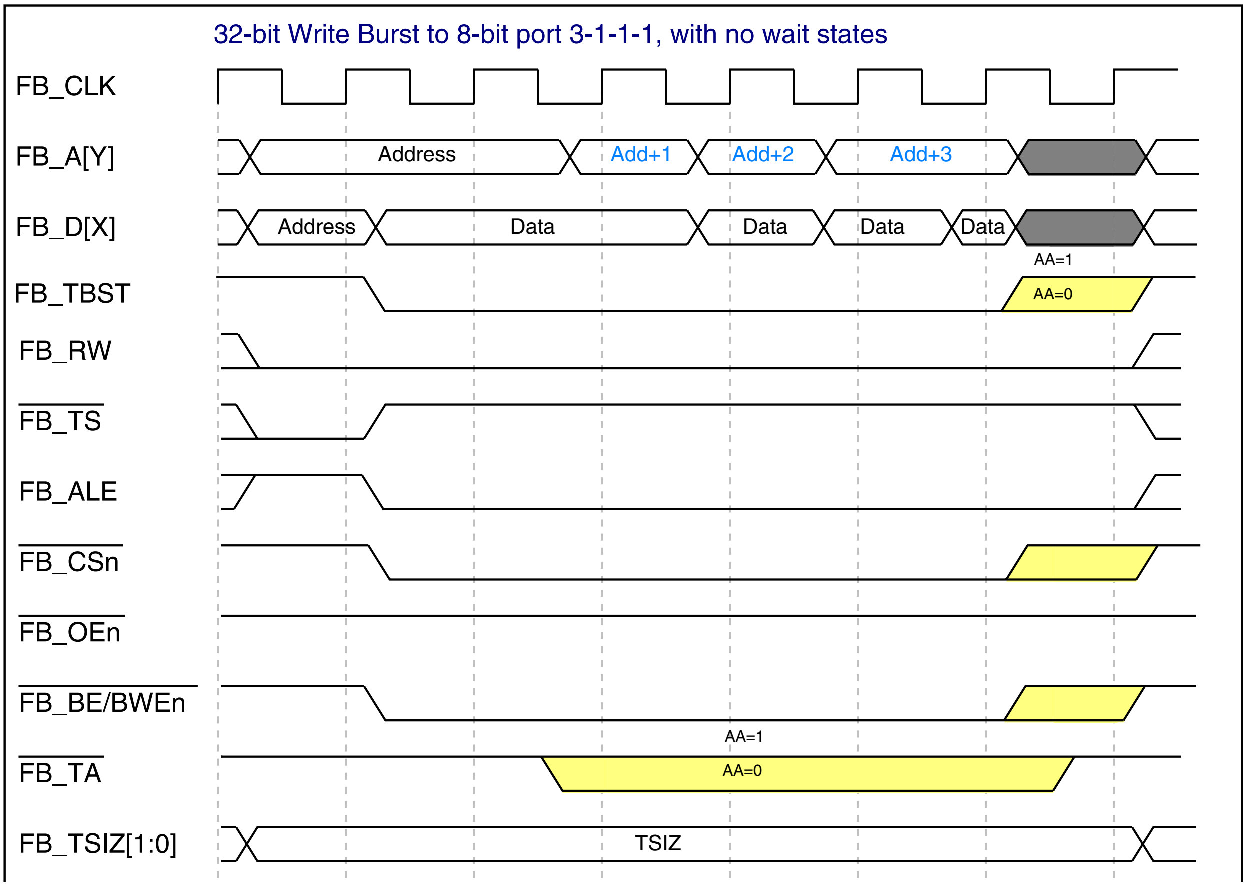


32.4.12.4 32-bit-Write burst to 8-Bit port 3-1-1-1 (no wait states)

The following figure shows a 32-bit write to an 8-bit external chip with burst enabled. The transfer results in a 4-beat burst and the data is driven on FB_AD[31:24]. The transfer size is driven at 32-bit (00b) throughout the bus cycle.

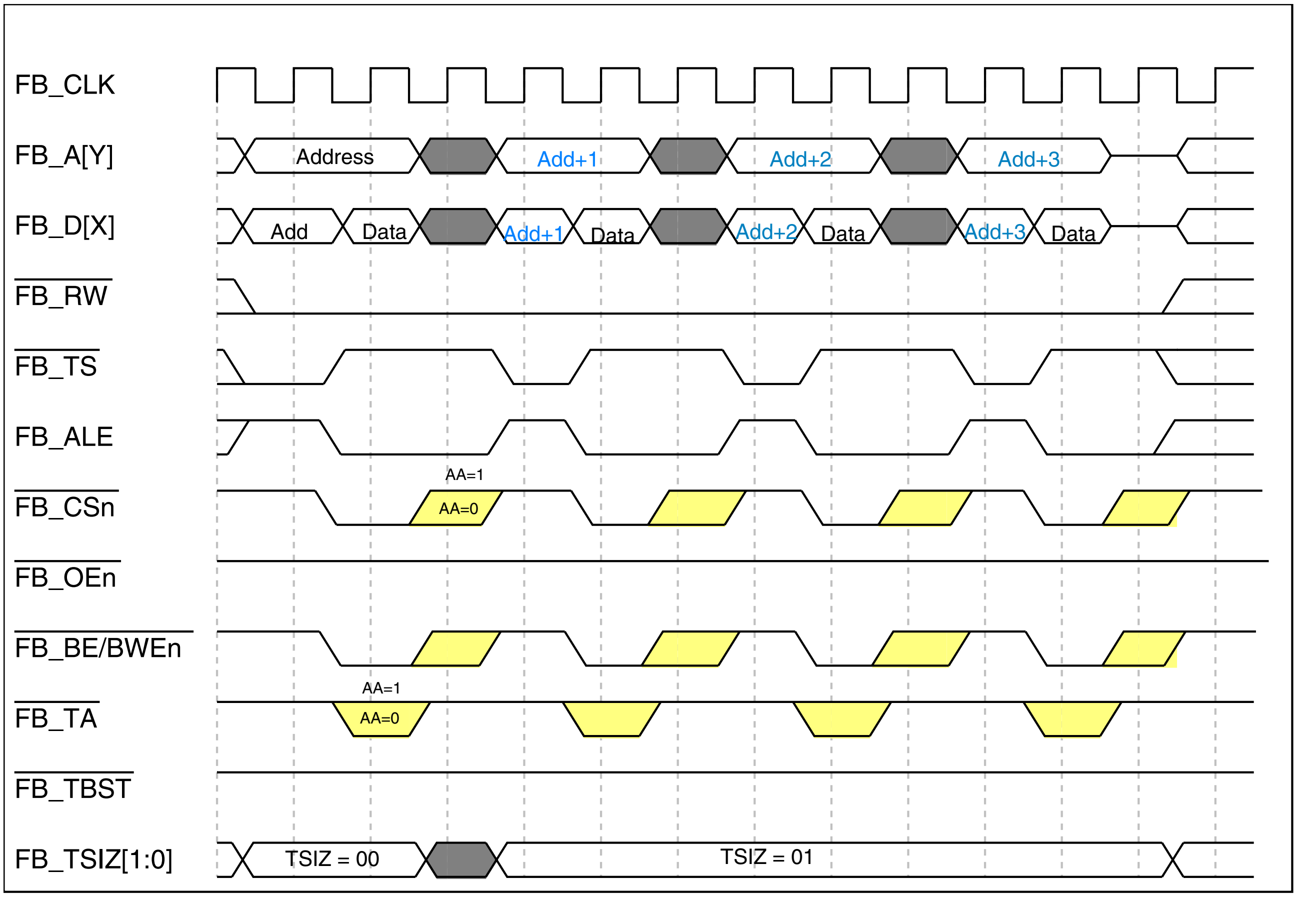
Note

The first beat of any write burst cycle has at least one wait state. If the bus cycle is programmed for zero wait states (CSCRn[WS] = 0b), one wait state is added. Otherwise, the programmed number of wait states are used.



32.4.12.5 32-bit-write burst-inhibited to 8-bit port (no wait states)

The following figure shows a 32-bit write to an 8-bit device with burst inhibited. The transfer results in four individual transfers. The transfer size is driven at 32-bit (00b) during the first transfer and at byte (01b) during the next three transfers.

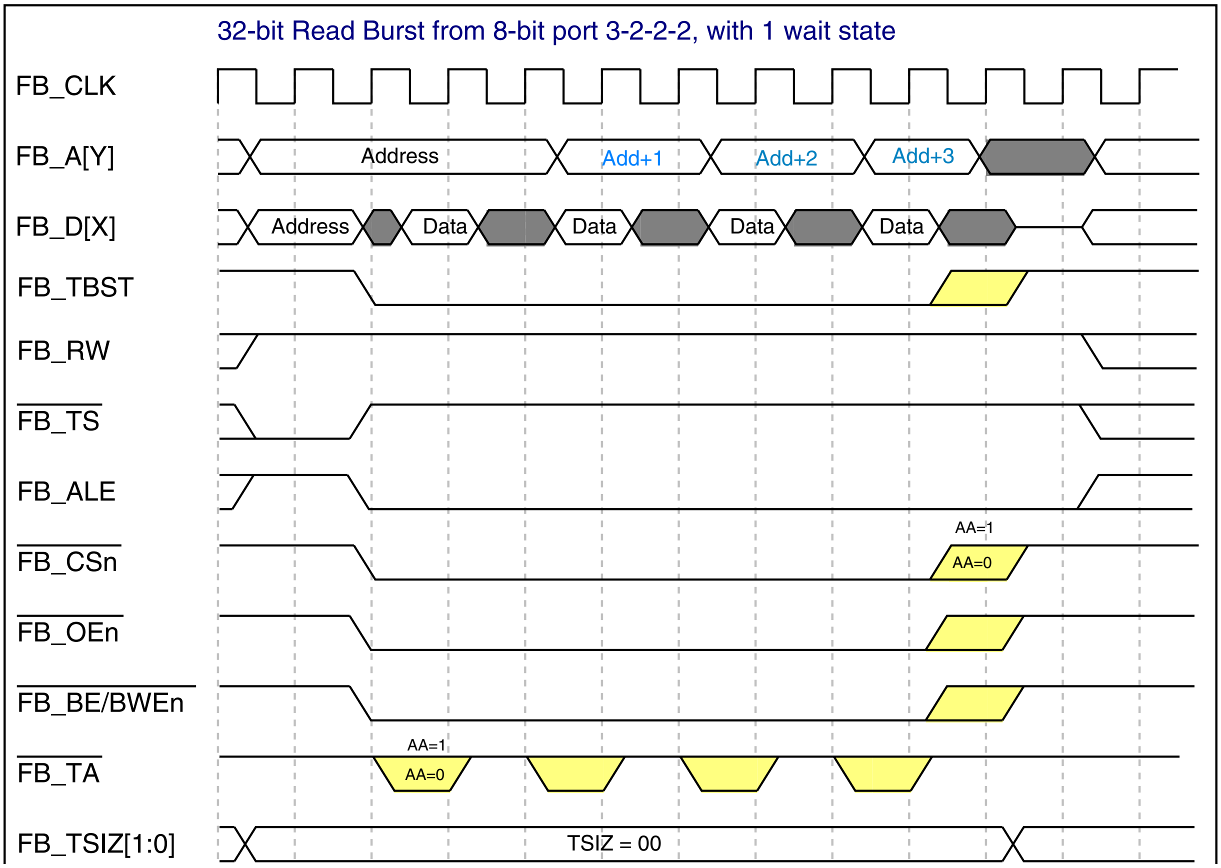


32.4.12.6 32-bit-read burst from 8-bit port 3-2-2-2 (one wait state)

The following figure illustrates another read burst transfer, but in this case a wait state is added between individual beats.

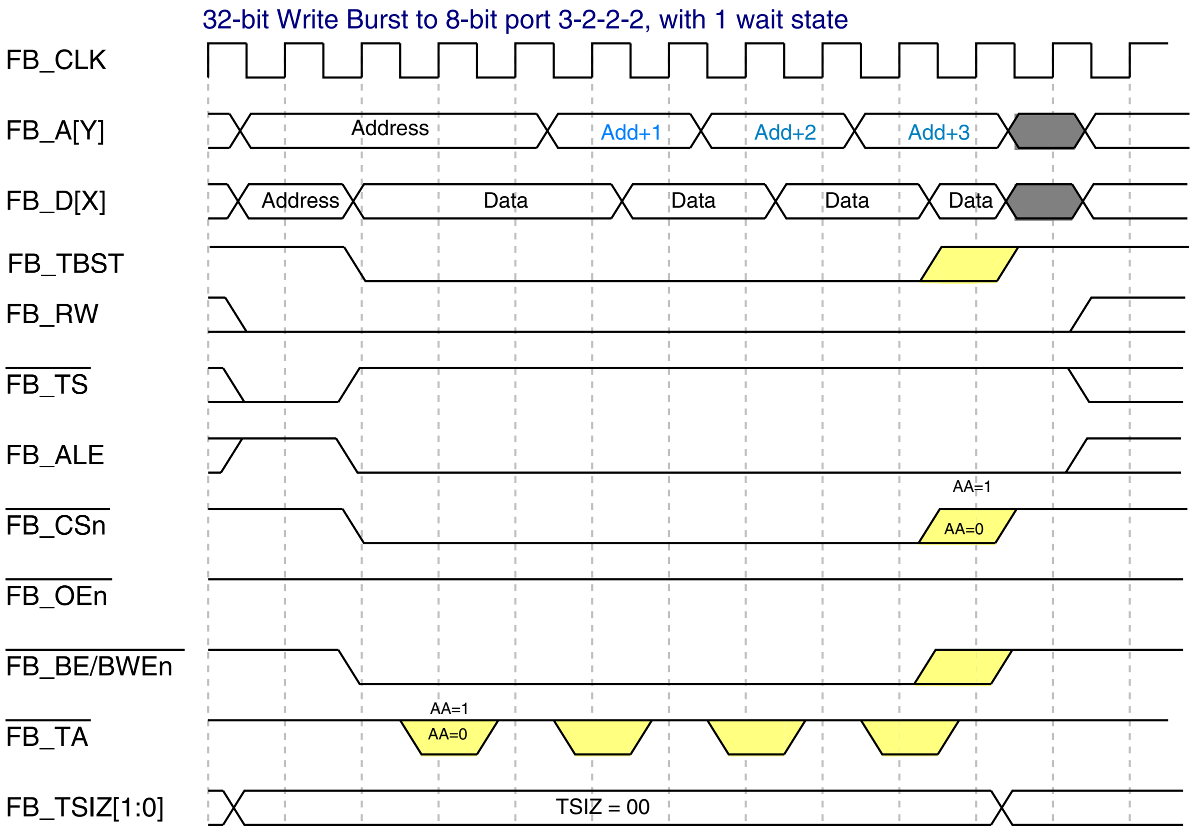
Note

CSCRn[WS] determines the number of wait states in the first beat. However, for subsequent beats, the CSCRn[WS] (or CSCRn[SWS] if CSCRn[SWSEN] = 1b) determines the number of wait states.



32.4.12.7 32-bit-write burst to 8-bit port 3-2-2-2 (one wait state)

The following figure illustrates a write burst transfer with one wait state.



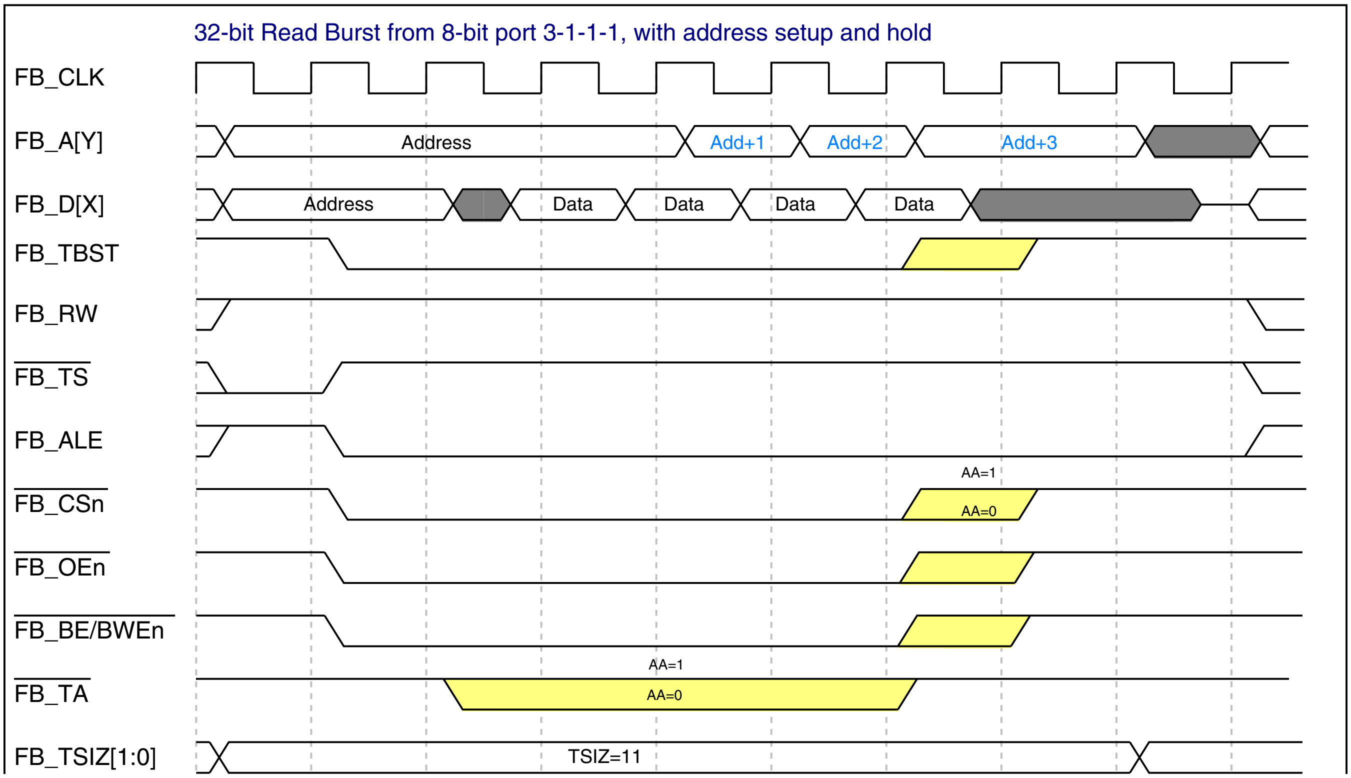
32.4.12.8 32-bit-read burst from 8-bit port 3-1-1-1 (address setup and hold)

If address setup and hold are used, only the first and last beat of the burst cycle are affected. The following figure shows a read cycle with one clock of address setup and address hold.

Note

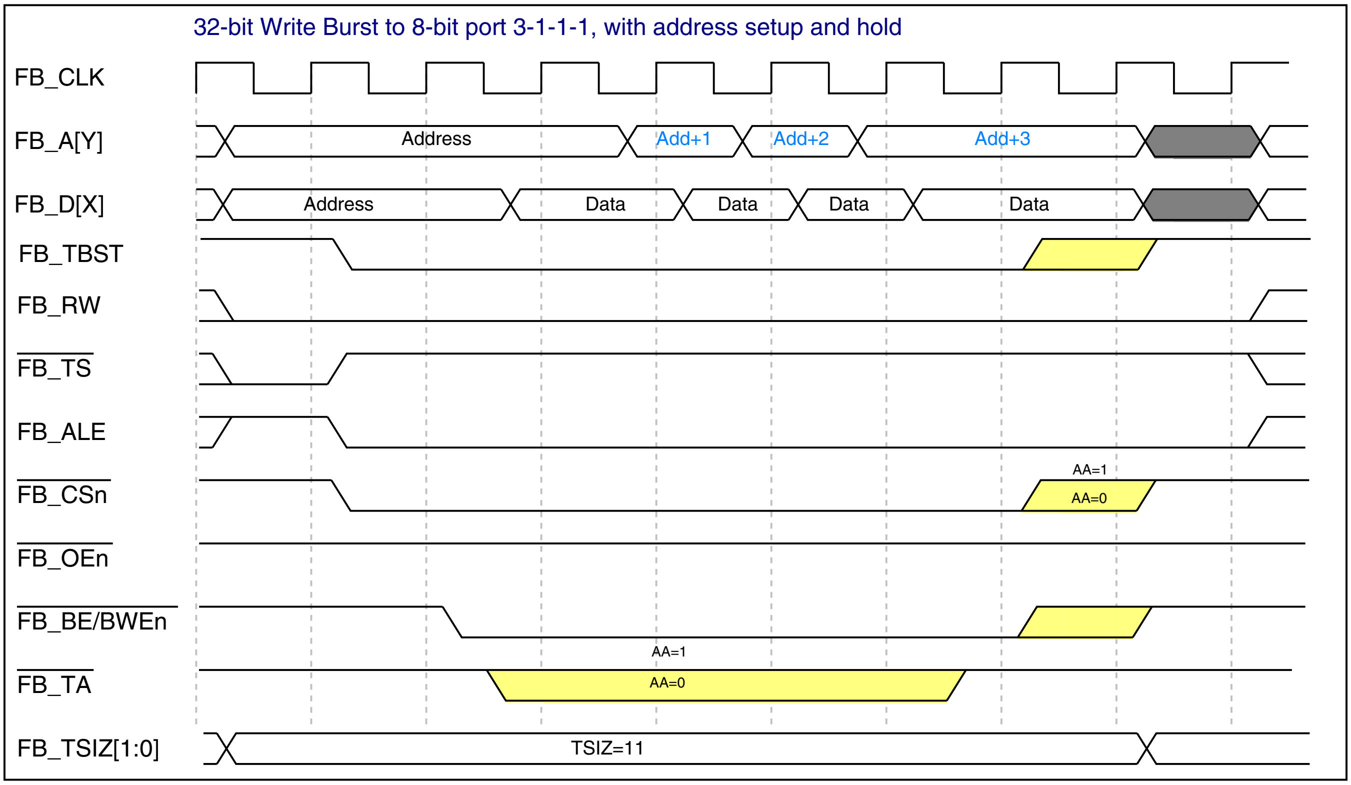
In non-multiplexed address/data mode, the address on FB_A increments only during internally-terminated burst cycles (CSCRn[AA] = 1b). The attached device must be able to account for this, or a wait state must be added. The first address is driven throughout the entire burst for externally-terminated cycles.

In multiplexed address/data mode, the address is driven on FB_AD only during the first cycle for internally- and externally-terminated cycles.



32.4.12.9 32-bit-write burst to 8-bit port 3-1-1-1 (address setup and hold)

The following figure shows a write cycle with one clock of address setup and address hold.



32.4.13 Extended Transfer Start/Address Latch Enable

The FB_TS/FB_ALE signal indicates that a bus transaction has begun and the address and attributes are valid. By default, the FB_TS/FB_ALE signal asserts for a single bus clock cycle. When CSCR*n*[EXTS] is set, the FB_TS/FB_ALE signal asserts and remain asserted until the first positive clock edge after FB_CS*n* asserts. See the following figure.

NOTE

When EXTS is set, CSCR*n*[WS] must be programmed to have at least one primary wait state.

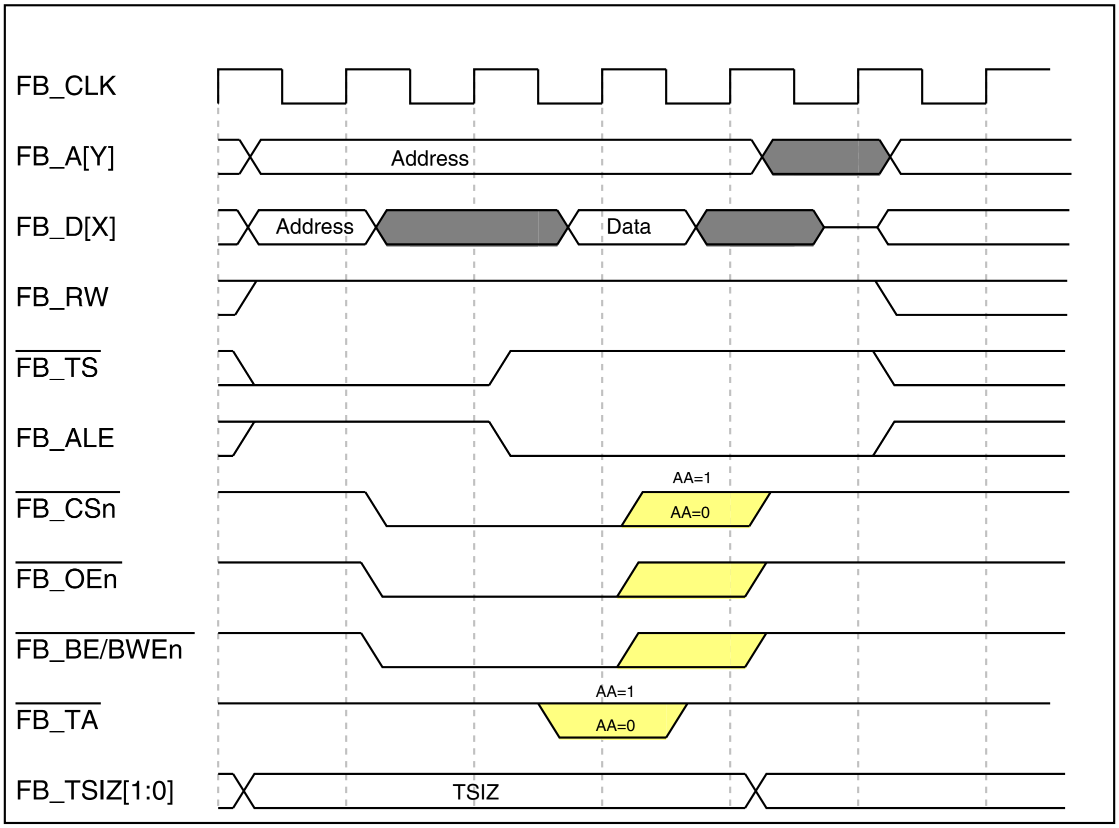


Figure 32-41. Read-Bus Cycle with CSCR*n*[EXTS] = 1 (One Wait State)

32.4.14 Bus errors

These types of accesses cause a transfer to terminate with a bus error:

- A write to a write-protected address range
- An access whose address is not in a range covered by a chip-select
- An access whose address is in a range covered by more than one chip-selects
- A write to a reserved address in the memory map
- A write to a reserved field in the CSPMCR
- Any FlexBus accesses when FlexBus is secure

If the auto-acknowledge feature is disabled (CSCR[AA] is 0) for an address that generates an error, the transfer can be terminated by asserting FB_TA. If the processor must manage a bus error differently, asserting an interrupt to the core along with FB_TA when the bus error occurs can invoke an interrupt handler.

The device can hang if FlexBus is configured for external termination and the CSPMCR is not configured for FB_TA.

32.5 Initialization/Application Information

32.5.1 Initializing a chip-select

To initialize a chip-select:

1. Write to the associated CSAR.
2. Write to the associated CSCR.
3. Write to the associated CSMR, including writing 1b to the Valid field (CSMRn[V]).

32.5.2 Reconfiguring a chip-select

To reconfigure a previously-used chip-select:

1. Invalidate the chip-select by writing 0b to the associated CSMR's Valid field (CSMRn[V]).
2. Write to the associated CSAR.
3. Write to the associated CSCR.
4. Write to the associated CSMR, including writing 1b to the Valid field (CSMRn[V]).

Chapter 33
Cyclic Redundancy Check (CRC)

33.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The cyclic redundancy check (CRC) module generates 16/32-bit CRC code for error detection.

The CRC module provides a programmable polynomial, WAS, and other parameters required to implement a 16-bit or 32-bit CRC standard.

The 16/32-bit code is calculated for 32 bits of data at a time.

33.1.1 Features

Features of the CRC module include:

- Hardware CRC generator circuit using a 16-bit or 32-bit programmable shift register
- Programmable initial seed value and polynomial
- Option for inversion of final CRC result
- 32-bit CPU register programming interface

33.1.2 Block diagram

The following is a block diagram of the CRC.

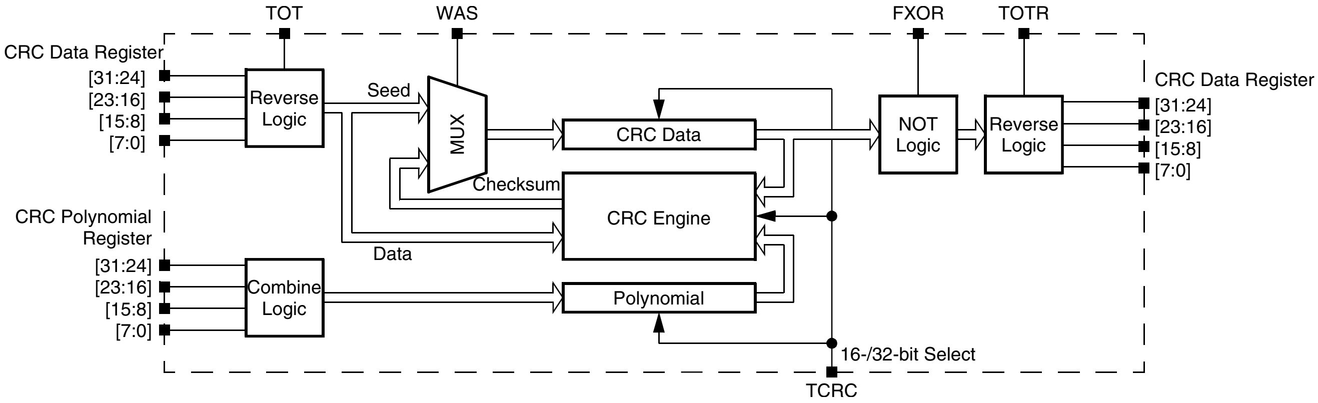


Figure 33-1. Programmable cyclic redundancy check (CRC) block diagram

33.1.3 Modes of operation

Various MCU modes affect the CRC module's functionality.

33.1.3.1 Run mode

This is the basic mode of operation.

33.1.3.2 Low-power modes (Wait or Stop)

Any CRC calculation in progress stops when the MCU enters a low-power mode that disables the module clock. It resumes after the clock is enabled or via the system reset for exiting the low-power mode. Clock gating for this module is dependent on the MCU.

33.2 Memory map and register descriptions

CRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_3000	CRC Data register (CRC_DATA)	32	R/W	FFFF_FFFFh	33.2.1/1381
4003_3004	CRC Polynomial register (CRC_GPOLY)	32	R/W	0000_1021h	33.2.2/1382
4003_3008	CRC Control register (CRC_CTRL)	32	R/W	0000_0000h	33.2.3/1382

33.2.1 CRC Data register (CRC_DATA)

The CRC Data register contains the value of the seed, data, and checksum. When CTRL[WAS] is set, any write to the data register is regarded as the seed value. When CTRL[WAS] is cleared, any write to the data register is regarded as data for general CRC computation.

In 16-bit CRC mode, the HU and HL fields are not used for programming the seed value, and reads of these fields return an indeterminate value. In 32-bit CRC mode, all fields are used for programming the seed value.

When programming data values, the values can be written 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous; with MSB of data value written first.

After all data values are written, the CRC result can be read from this data register. In 16-bit CRC mode, the CRC result is available in the LU and LL fields. In 32-bit CRC mode, all fields contain the result. Reads of this register at any time return the intermediate CRC value, provided the CRC module is configured.

Address: 4003_3000h base + 0h offset = 4003_3000h

Bit	31–24	23–16	15–8	7–0
R/W	HU	HL	LU	LL
Reset	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

CRC_DATA field descriptions

Field	Description
31–24 HU	CRC High Upper Byte In 16-bit CRC mode (CTRL[TCRC] is 0), this field is not used for programming a seed value. In 32-bit CRC mode (CTRL[TCRC] is 1), values written to this field are part of the seed value when CTRL[WAS] is 1. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation in both 16-bit and 32-bit CRC modes.
23–16 HL	CRC High Lower Byte In 16-bit CRC mode (CTRL[TCRC] is 0), this field is not used for programming a seed value. In 32-bit CRC mode (CTRL[TCRC] is 1), values written to this field are part of the seed value when CTRL[WAS] is 1. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation in both 16-bit and 32-bit CRC modes.
15–8 LU	CRC Low Upper Byte When CTRL[WAS] is 1, values written to this field are part of the seed value. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation.
7–0 LL	CRC Low Lower Byte When CTRL[WAS] is 1, values written to this field are part of the seed value. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation.

33.2.2 CRC Polynomial register (CRC_GPOLY)

This register contains the value of the polynomial for the CRC calculation. The HIGH field contains the upper 16 bits of the CRC polynomial, which are used only in 32-bit CRC mode. Writes to the HIGH field are ignored in 16-bit CRC mode. The LOW field contains the lower 16 bits of the CRC polynomial, which are used in both 16- and 32-bit CRC modes.

Address: 4003_3000h base + 4h offset = 4003_3004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HIGH																LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1

CRC_GPOLY field descriptions

Field	Description
31–16 HIGH	High Polynominal Half-word Writable and readable in 32-bit CRC mode (CTRL[TCRC] is 1). This field is not writable in 16-bit CRC mode (CTRL[TCRC] is 0).
15–0 LOW	Low Polynominal Half-word Writable and readable in both 32-bit and 16-bit CRC modes.

33.2.3 CRC Control register (CRC_CTRL)

This register controls the configuration and working of the CRC module. Appropriate bits must be set before starting a new CRC calculation. A new CRC calculation is initialized by asserting CTRL[WAS] and then writing the seed into the CRC data register.

Address: 4003_3000h base + 8h offset = 4003_3008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0		0	FXOR	WAS	TCRC	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRC_CTRL field descriptions

Field	Description
31–30 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
29–28 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 FXOR	Complement Read Of CRC Data Register Some CRC protocols require the final checksum to be XORed with 0xFFFFFFFF or 0xFFFF. Asserting this bit enables on the fly complementing of read data. 0 No XOR on reading. 1 Invert or complement the read value of the CRC Data register.
25 WAS	Write CRC Data Register As Seed When asserted, a value written to the CRC data register is considered a seed value. When deasserted, a value written to the CRC data register is taken as data for CRC computation. 0 Writes to the CRC data register are data values. 1 Writes to the CRC data register are seed values.
24 TCRC	Width of CRC protocol. 0 16-bit CRC protocol. 1 32-bit CRC protocol.
23–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

33.3 Functional description

33.3.1 CRC initialization/reinitialization

To enable the CRC calculation, the user must program CRC_CTRL[WAS], CRC_GPOLY, and CRC result inversion in the applicable registers. Asserting CRC_CTRL[WAS] enables the programming of the seed value into the CRC_DATA register.

After a completed CRC calculation, reasserting CRC_CTRL[WAS] and programming a seed, whether the value is new or a previously used seed value, reinitialize the CRC module for a new CRC computation. All other parameters must be set before programming the seed value and subsequent data values.

33.3.2 CRC calculations

In 16-bit and 32-bit CRC modes, data values can be programmed 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous. Noncontiguous bytes can lead to an incorrect CRC computation.

33.3.2.1 16-bit CRC

To compute a 16-bit CRC:

1. Clear CRC_CTRL[TCRC] to enable 16-bit CRC mode.
2. Write a 16-bit polynomial to the CRC_GPOLY[LOW] field. The CRC_GPOLY[HIGH] field is not usable in 16-bit CRC mode.
3. Set CRC_CTRL[WAS] to program the seed value.
4. Write a 16-bit seed to CRC[LU:LL]. CRC_DATA[HU:HL] are not used.
5. Clear CRC_CTRL[WAS] to start writing data values.
6. Write data values into CRC[LU:LL]. A CRC is computed on every data value write, and the intermediate CRC result is stored back into CRC[LU:LL].
7. When all values have been written, read the final CRC result from CRC[LU:LL].

33.3.2.2 32-bit CRC

To compute a 32-bit CRC:

1. Set CRC_CTRL[TCRC] to enable 32-bit CRC mode.
2. Write a 32-bit polynomial to CRC_GPOLY[HIGH:LOW].
3. Set CRC_CTRL[WAS] to program the seed value.
4. Write a 32-bit seed to CRC[LL].
5. Clear CRC_CTRL[WAS] to start writing data values.
6. Write data values into CRC[LL]. A CRC is computed on every data value write, and the intermediate CRC result is stored back into CRC[LL].
7. When all values have been written, read the final CRC result from CRC[LL]. The CRC is calculated bytewise, and two clocks are required to complete one CRC calculation.

33.3.3 CRC result complement

When CTRL[FXOR] is set, the checksum is complemented. The CRC result complement function outputs the complement of the checksum value stored in the CRC data register every time the CRC data register is read. When CTRL[FXOR] is cleared, reading the CRC data register accesses the raw checksum value.

Chapter 34
LPDDR2/DDR3 SDRAM Memory Controller (DDRMC)

34.1 Introduction

The memory controller supports high performance applications for 16-bit or 8-bit DDR2, or LPDDR SDRAM memories.

The features of this Memory Controller include:

- Supports interfacing to LPDDR2 and DDR3 memory types.
- Fully pipelined command, read and write data interfaces to the memory controller.
- Advanced bank look-ahead features for high memory throughput.
- Front-end interface to 2 standard AXI ports.
- A programmable register interface to control memory parameters and protocols including auto pre-charge.
- Full initialization of memory on memory controller reset.
- ECC functionality with single bit and double bit error reporting and automatic correction of single bit error events. 10 bit memory interface required for ECC (8 bit user data + 2 bits ECC). Programmable removal of ECC storage.
- Programmable memory datapath size of full memory data (16 bit) width or half memory data (8 bit) width.
- Clock frequencies from 100 MHz to 400 MHz supported.
- Back-end interface to a PHY.
- Integrates support for DDR pad calibration logic calibration in both software mode and hardware mode. Software calibration is only for debug.

34.2 Block diagram

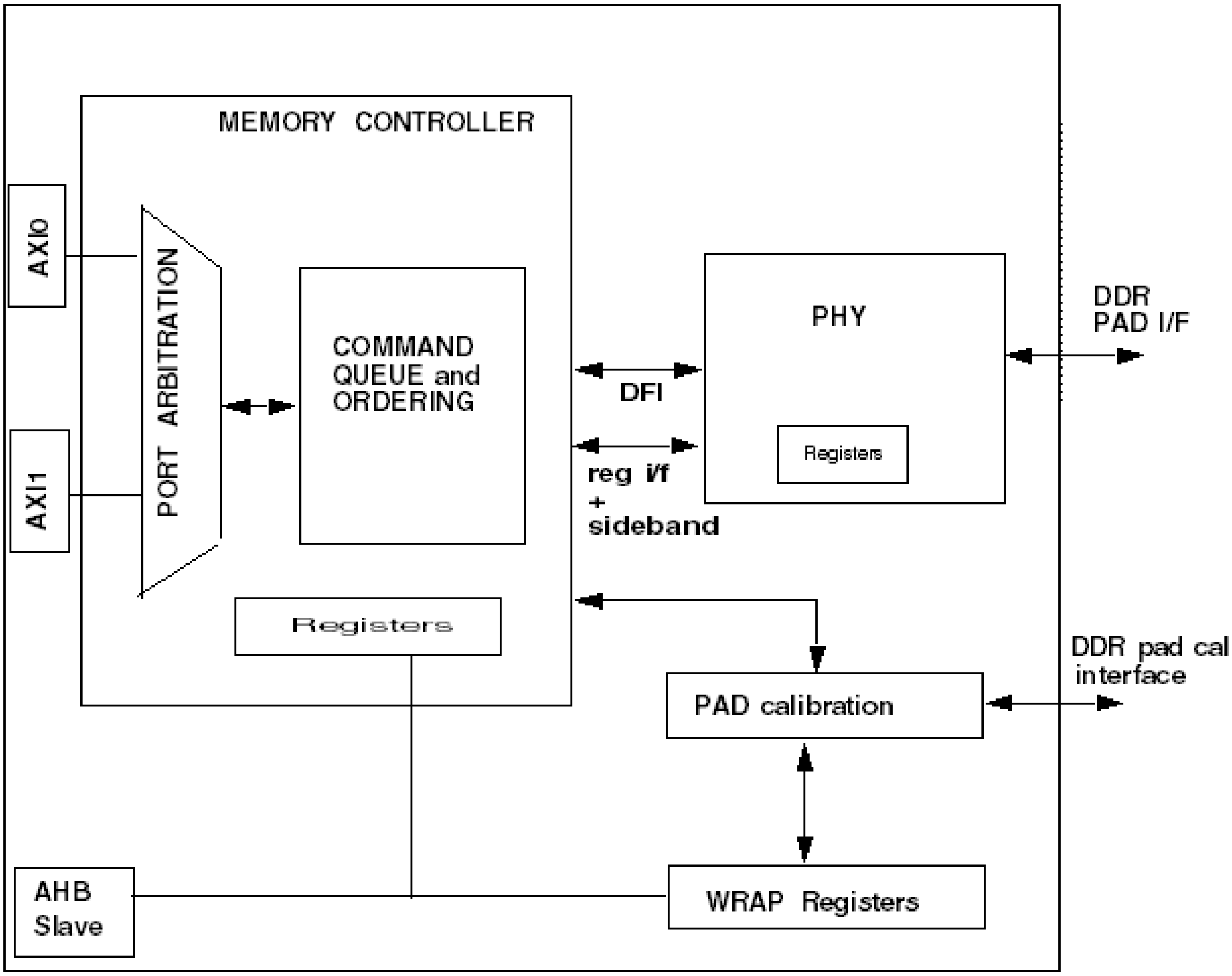


Figure 34-1. DDRMC Block Diagram

34.3 Modes of Operation

The following operating modes are available:

- LPDDR2
- DDR3

34.3.1 Low Power Modes

For details see Low Power Operation.

34.4 Signal Description

The following table lists the memory controller's external memory interface signals.

Table 34-1. Signal Properties

Name	Function	I/O	Reset
DDR_A[10:0] DDR_A[14:0]	Selects the column when DDR_CAS is asserted Selects the row when DDR_RAS is asserted.	O	0000
DDR_BA[2:0]	Selects 1 of 8 SDRAM Banks when DDR_RAS is asserted (precharge or active) or when DDR_CAS is asserted (read or write). 000 - Bank1 001 - Band2 ---------- 111 - Bank8	O	0
DDR_CAS	Column address select	O	1
DDR_CKE	Clock enable. Asserts when the clock is valid	O	0
DDR_CLK DDR_CLK	Differential clock	O	0 1
DDR_CS	Chip select	O	1
DDR_DQ[15:0]	Write or read data from the SDRAM	I/O	—
DDR_DM	Data mask: 00 No mask 01 Low byte mask enable 10 High byte mask enable 11 Low and hign byte mask enable	O	11
DDR_DQS[1:0]	Differential data strobe **Note:** Pull down is active in LPDDR2 mode only.	I/O	—
DDR_ODT	On-die-termination	O	0
DDR_RAS	Row address select	O	1
DDR_WE	Write enable	O	1
DDR_VREF	Voltage supply reference	—	—
DDR_RESET	Reset signal	O	—

34.4.1 Detailed Signal Descriptions

The following table describes the external signals in more detail with timing to DDR_CLK and how the signals interact with each other.

Table 34-2. Detailed Signal Descriptions

Signal	I/O		Description
DDR_A[15:0]	O		Provides the row address for ACTIVE commands, and the column address and AUTO PRECHARGE bit for READ/WRITE commands, to select one location out of the memory array in the respective bank. A10 is sampled during a precharge command to determine whether the PRECHARGE applies to one bank (A10 LOW) or all banks (A10 HIGH). If only one bank is to be precharged, the bank is selected by BA0, BA1 and BA2. The address outputs also provide the op-code during a MODE REGISTER SET command. BA0, BA1 and BA2 define which mode register is loaded during the MODE REGISTER SET (MRS or EMRS's)
		State Meaning	Please seeTable 34-4 for the LPDDR2 SDRAM commands.
		Timing	Assertion/Negation— Occurs synchronously with DDR_CLK
DDR_BA[2:0]	O		BA0, BA1 and BA2 define which 1 of 8 banks an ACTIVE, READ, WRITE, or PRECHARGE command is being applied to.
		Timing	Assertion/Negation— Occurs synchronously with DDR_CLK
$\overline{\text{DDR\_CAS}}$	O		Command input. Along with $\overline{\text{DDR\_CS}}$, $\overline{\text{DDR\_RAS}}$, and $\overline{\text{DDR\_WE}}$ defines the current command.
		State Meaning	Please seeTable 34-3 for the DDR3 SDRAM commands.
		Timing	Assertion/Negation— Occurs synchronously with DDR_CLK
DDR_CKE	O		CKE must be maintained high throughout READ and WRITE accesses. Input buffers, excluding DDR_CLK, $\overline{\text{DDR\_CLK}}$, and DDR_CKE, are disabled during POWER DOWN or SELF REFRESH.
		State Meaning	Asserted— Activates internal clock signals and device input buffers and output drivers. Negated—Deactivates internal clock signals and device input buffers and output drivers.
		Timing	Assertion — Asynchronous for SELF-REFRESH exit and for output disable Negation— Occurs synchronously with DDR_CLK
DDR_CLK $\overline{\text{DDR\_CLK}}$	O		DDR_CLK and $\overline{\text{DDR\_CLK}}$ are differential clock outputs. All address and control output signals are sent on the crossing of the positive edge of DDR_CLK and the negative edge of $\overline{\text{DDR\_CLK}}$. Output data is referenced to the crossing of DDR_CLK and $\overline{\text{DDR\_CLK}}$ (both directions of crossing).
		Timing	Command signals are synchronously with the rising edge of this clock. The data signals can change on both the rising and falling edge of the clock.
$\overline{\text{DDR\_CS}}$	O		$\overline{\text{DDR\_CS}}$ provides for external chip selection on systems with multiple chips. $\overline{\text{DDR\_CS}}$ is considered part of the command code.
		State Meaning	Asserted— Commands for the selected chip will occur Negated—All commands are masked.
		Timing	Assertion/Negation— Occurs synchronously with DDR_CLK
DDR_D[15:0]	I/O		Data bus
		Timing	Assertion/Negation— Occurs on both crossing of DDR_CLK and $\overline{\text{DDR\_CLK}}$ on write command. Synchronous with DDR_DQS input on read command.

Table continues on the next page...

Table 34-2. Detailed Signal Descriptions (continued)

Signal	I/O	Description	
DDR_DM[1:0]	O	Output mask signal for write data. During Reads, DDR_DM may be driven high, low, or floating.	
		State Meaning	Asserted— Data is written to SDRAM Negation— Data is masked
		Timing	Assertion/Negation— Occurs on both crossing of DDR_CLK and DDR_CLK.
DDR_DQS[1:0]	I/O	Edge-aligned with read data, centered in write data. Used to capture data.	
		State Meaning	Asserted— Similar to a clock signal, the edges are more important than being asserted or negated.
		Timing	Assertion/Negation—Occurs on both crossing of DDR_CLK and DDR_CLK on write command. Asynchronous with DDR_CLK and DDR_CLK on read command.
DDR_ODT	O	DDR_ODT enables termination resistance internal to the DDR2 SDRAM.	
		State Meaning	Asserted— Enable termination resistance Negation— Disable termination resistance Please see Table 34-3 for DDR3 SDRAM commands.
		Timing	Assertion/Negation— Occurs synchronously with DDR_CLK
DDR_RAS	O	Command input. Along with DDR_CS, DDR_CAS, and DDR_WE defines the current command.	
		State Meaning	Please see Table 34-3 for DDR3 SDRAM commands.
		Timing	Assertion/Negation— Occurs synchronously with DDR_CLK.
DDR_WE	O	Command input. Along with DDR_CS, DDR_CAS, and DDR_RAS defines the current command.	
		State Meaning	Please see Table 34-3 for DDR3 SDRAM commands.
		Timing	Assertion/Negation— Occurs synchronously with DDR_CLK. Refer device datasheet for the PU/PD details.
DDR_VREF	—	SDRAM reference voltage. Reference voltage for differential I/O pad cells should be half the voltage of the memory used in the system. For example, 1.5 V DDR3 results in an DDR_VREF of 0.75 V and 1.2 V LPDDR2 results in an DDR_VREF of 0.6 V. See the device's datasheet for the voltages and tolerances for the various memory modes.	
DDR_RESET	O	DDR3 memory reset signal.	
		State Meaning	Assertion: Memory is in Reset Negation: Memory is in normal operating mode
		Timing	ASSERTS asynchronously and de-assertion is synchronour to DDR_CLK per JEDEC timings

Table 34-3. DDR3 SDRAM Command truth table

Function	Abbreviation	CKE		CS#	RAS#	CAS#	WE#	BA0-BA2	A13-A15	A12-BC#	A10-AP	A0-A9,A11	Notes
		Previous Cycle	Current Cycle										
Mode Register Set	MRS	H	H	L	L	L	L	BA	OP Code				

Table continues on the next page...

Table 34-3. DDR3 SDRAM Command truth table (continued)

Function	Abbreviation	CKE		CS#	RAS#	CAS#	WE#	BA0-BA2	A13-A15	A12-BC#	A10-AP	A0-A9,A11	Notes
		Previous Cycle	Current Cycle										
Refresh	REF	H	H	L	L	L	H	V	V	V	V	V	
Self Refresh Entry	SRE	H	L	L	L	L	H	V	V	V	V	V	7,9,12
Self Refresh Exit	SRX	L	H	H	X	X	X	X	X	X	X	X	7,8,9,12
				L	H	H	H	V	V	V	V	V	
Single Bank Precharge	PRE	H	H	L	L	H	L	BA	V	V	L	V	
Precharge all Banks	PREA	H	H	L	L	H	L	V	V	V	H	V	
Bank Activate	ACT	H	H	L	L	H	H	BA	Row Address (RA)				
Write(Fixed BL8 or BC4)	WR	H	H	L	H	L	L	BA	RFU	V	L	CA	
Write (BC4, on the Fly)	WRS4	H	H	L	H	L	L	BA	RFU	L	L	CA	
Write(BL8, on the Fly)	WRSS	H	H	L	H	L	L	BA	RFU	H	L	CA	
Write with Auto Precharge (Fixed BL8 or BC4)	WRA	H	H	L	H	L	L	BA	RFU	V	H	CA	
Write with Auto Precharge (BC4, on the Fly)	WRAS4	H	H	L	H	L	L	BA	RFU	L	H	CA	

Table continues on the next page...

Table 34-3. DDR3 SDRAM Command truth table (continued)

Function	Abbreviation	CKE Previous Cycle	CKE Current Cycle	CS#	RAS#	CAS#	WE#	BA0-BA2	A13-A15	A12-BC#	A10-AP	A0-A9,A11	Notes
Write with Auto Precharge (BL8,on the Fly)	WRAS8	H	H	L	H	L	L	BA	RFU	H	H	CA	
Read (Fixed BL8 or BC4)	RD	H	H	L	H	L	H	BA	RFU	V	L	CA	
Read (BC4, on the Fly	RDS4	H	H	L	H	L	H	BA	RFU	L	L	CA	
Read (BL8, on the Fry)	RDS8	H	H	L	H	L	H	BA	RFU	H	L	CA	
Read with Auto Precharge (Fixed BL8 or BC4)	RDA	H	H	L	H	L	H	BA	RFU	V	H	CA	
Read with Auto Precharge (BC4, on the Fly)	RDAS4	H	H	L	H	L	H	BA	RFU	L	H	CA	
Read with Auto Precharge (Fixed BL8, on the Fly)	RDASS	H	H	L	H	L	H	BA	RFU	H	H	CA	
No Operation	NOP	H	H	L	H	H	H	V	V	V	V	V	10

Table continues on the next page...

Table 34-3. DDR3 SDRAM Command truth table (continued)

Function	Abbreviation	CKE		CS#	RAS#	CAS#	WE#	BA0-BA2	A13-A15	A12-BC#	A10-AP	A0-A9,A11	Notes
		Previous Cycle	Current Cycle										
Device Deselected	DES	H	H	H	X	X	X	X	X	X	X	X	11
Power Down Entry	PDE	H	L	L	H	H	H	V	V	V	V	V	6,12
				H	X	X	X	X	X	X	X	X	
Power Down Exit	X	L	H	L	H	H	H	V	V	V	V	V	6,12
				H	X	X	X	X	X	X	X	X	
ZQ Calibration Long	L	H	H	L	H	H	L	X	X	X	H	X	
ZQ Calibration Short	ZQCS	H	H	L	H	H	L	X	X	X	L	X	

Note

1. CKE is HIGH for all commands shown except SELF REFRESH.
2. BA0--BA1 select either the Base or the Extended Mode Register (BA0 = 0, BA1 = 0 selects Mode Register; BA0 = 1, BA1 = 0 selects Extended Mode Register; other combinations of BA0--BA1 are reserved; A0--A13 provide the op--code to be written to the selected Mode Register.
3. BA0--BA1 provide bank address and A0--A13 provide row address.
4. BA0--BA1 provide bank address; A0--Ai provide column address; A10 HIGH enables the auto precharge feature (non-persistent), A10 LOW disables the auto precharge feature.
5. A10 LOW: BA0--BA1 determine which bank is precharged. A10 HIGH: all banks are precharged and BA0--BA1 are "Don't Care."
6. This command is AUTO REFRESH if CKE is HIGH; SELF REFRESH if CKE is LOW.

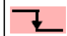
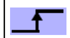
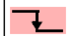
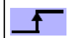
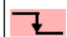
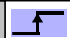
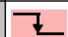
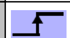
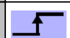
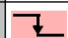
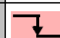
7. Internal refresh counter controls row addressing; all inputs and I/Os are "Don't Care" except for CKE.
8. Applies only to read bursts with autoprecharge disabled; this command is undefined (and should not be used) for read bursts with autoprecharge enabled, and for write bursts.
9. DESELECT and NOP are functionally interchangeable.
10. Used to mask write data, provided coincident with the corresponding data.
11. Operation or timing that is not specified is illegal and after such an event, in order to guarantee proper operation, the DRAM must be powered down and then restarted through the specified initialization sequence before normal operation can continue.
12. VREF must be maintained during Self Refresh operation.

Table 34-4. LPDDR2 SDRAM Command truth table

	SDR Command Pins			DDR CA pins (10)										
SDRAM Command	CKE		CS_N	CA0	CA1	CA2	CA3	CA4	CA5	CA6	CA7	CA8	CA9	CK EDGE
	CK_t(n-1)	CK_t(n)												
MRW	H	H	L	L	L	L	L	MA0	UA1	MA2	MA3	MA4	MA5	
				MA6	MA7	P0	OP1	OP2	OP3	OP4	OP5	OP6	OP7	
MRR	H	H	L	L	L	L	H	MA0	MA1	MA2	MA3	MA4	MA5	
				MA6	MA7	X								
Refresh (per bank)[11]	H	H	L	L	L	H	L	X						
				X										
Refresh (all bank)	H	H	L	L	L	H	H	X						
				X										
Enter Self Refresh	H	L	L	L	L	H	X							
				X										
Activate [bank]	H	H	L	L	H	R8/ a15	R9/ a16	R10/ a17	R11/ a18	R12/ a19	BA0	BA1	BA2	
				R0/ a5	R1/ a5	R2/ a7	R3/ a8	R4/ a9	R5/ a10	R6/ a11	R7/ a12	R13/ a13	R14/ a14	

Table continues on the next page...

Table 34-4. LPDDR2 SDRAM Command truth table (continued)

	SDR Command Pins			DDR CA pins (10)										
SDRAM Command	CKE		CS_N	CA0	CA1	CA2	CA3	CA4	CA5	CA6	CA7	CA8	CA9	CK EDGE
	CK_t(n-1)	CK_t(n)												
Write (bank)	H	H	L	H	L	L	RFU	RFU	C1	C2	BA0	BA1	BA2	
				AP3, 4	C3	C4	C5	C6	C7	C8	C9	C10	C11	
Read (bank)	H	H	L	H	L	H	RFU	RFU	C1	C2	BA0	BA1	BA2	
				AP3, 4	C3	C4	C5	C6	07	C8	C9	C10	C11	
Precharge (bank)	H	H	L	H	H	L	H	AB/ a30	X/ a31	X/ a32	BA0	BA1	BA2	
				X/ a20	X/ a21	X/ a22	X/ a23	X/ a24	X/ a25	X/ a26	X/ a27	X/ a28	X/ a29	
BST	H	H	L	H	H	L	L	X						
				X										
Enter Deep Power Down	H	L	L	H	H	L	X							
				X										
NOP	H	H	L	H	H	H	X							
				X										
Maintain PD, SREF, DPD (NOP)	L	L	L	H	H	H	X							
				X										
NOP	H	H	H	X										
				X										
Maintain PD, SREF, DPD (NOP)	L	L	H	X										
				X										
Enter Power Down	H	L	H	X										
				X										
Exit PD, SREF, DPD	L	H	H	X										
				X										

Note

1. All LPDDR2 commands are defined by states of CS_n, CA0, CA1, CA2, CA3, and CKE at the rising edge of the clock. CA refers to command/address slice.
2. For LPDDR2 SDRAM, Bank addresses BA0, BA1, BA2 (BA) determine which bank is to be operated upon.

3. AP is significant only to SDRAM.
4. AP "high" during a READ or WRITE command indicates that an auto-precharge will occur to the bank associated with the READ or WRITE command.
5. "X" means "H or L (but a defined logic level)"
6. Self refresh exit and Deep Power Down exit are asynchronous.
7. VREF must be between 0 and VDDQ during Self Refresh and Deep Power Down operation.
8. CAxr refers to command/address bit "x" on the rising edge of clock.
9. CAxf refers to command/address bit "x" on the falling edge of clock.
10. CS_n and CKE are sampled at the rising edge of clock.
11. Per Bank Refresh is only allowed in devices with 8 banks.
12. The least-significant column address C0 is not transmitted on the CA bus, and is implied to be zero.

34.5 Memory map and register description

NOTE

Register read or write for any register in the DDR controller requires the DDRMC clock to be configured from CCM_CCSR [DDRC_CLK_SEL] in addition to enabling the clock from the clock gating register: CCM_CCGR6[CG110].

DDRMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E000	Control Register 0 (DDRMC_CR00)	32	R/W	2041_0000h	34.5.1/1407
400A_E004	Control Register 1 (DDRMC_CR01)	32	R	0002_0D10h	34.5.2/1408
400A_E008	Control Register 2 (DDRMC_CR02)	32	R/W	0000_0000h	34.5.3/1409

Table continues on the next page...

DDRMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E00C	Control Register 3 (DDRMC_CR03)	32	R/W	0000_0000h	34.5.4/1409
400A_E010	Control Register 4 (DDRMC_CR04)	32	R/W	0000_0000h	34.5.5/1410
400A_E014	Control Register 5 (DDRMC_CR05)	32	R/W	0000_0000h	34.5.6/1410
400A_E018	Control Register 6 (DDRMC_CR06)	32	R (reads 0)	0000_0000h	34.5.7/1410
400A_E01C	Control Register 7 (DDRMC_CR07)	32	R (reads 0)	0000_0000h	34.5.8/1411
400A_E020	Control Register 8 (DDRMC_CR08)	32	R (reads 0)	0000_0000h	34.5.9/1411
400A_E024	Control Register 9 (DDRMC_CR09)	32	R/W	0000_0000h	34.5.10/ 1412
400A_E028	Control Register 10 (DDRMC_CR10)	32	R/W	0000_0000h	34.5.11/ 1412
400A_E02C	Control Register 11 (DDRMC_CR11)	32	R/W	0000_0000h	34.5.12/ 1413
400A_E030	Control Register 12 (DDRMC_CR12)	32	R/W	0000_0000h	34.5.13/ 1413
400A_E034	Control Register 13 (DDRMC_CR13)	32	R/W	0000_0000h	34.5.14/ 1414
400A_E038	Control Register 14 (DDRMC_CR14)	32	R/W	0000_0000h	34.5.15/ 1414
400A_E03C	Control Register 15 (DDRMC_CR15)	32	R (reads 0)	0000_0000h	34.5.16/ 1415
400A_E040	Control Register 16 (DDRMC_CR16)	32	R/W	0000_0000h	34.5.17/ 1415
400A_E044	Control Register 17 (DDRMC_CR17)	32	R/W	0000_0000h	34.5.18/ 1416
400A_E048	Control Register 18 (DDRMC_CR18)	32	R/W	0000_0000h	34.5.19/ 1417
400A_E04C	Control Register 19 (DDRMC_CR19)	32	R (reads 0)	0000_0000h	34.5.20/ 1417
400A_E050	Control Register 20 (DDRMC_CR20)	32	R/W	0000_0000h	34.5.21/ 1418
400A_E054	Control Register 21 (DDRMC_CR21)	32	R/W	0000_0000h	34.5.22/ 1419
400A_E058	Control Register 22 (DDRMC_CR22)	32	R/W	0000_0000h	34.5.23/ 1420
400A_E05C	Control register 23 (DDRMC_CR23)	32	R/W	0100_0000h	34.5.24/ 1420
400A_E060	Control Register 24 (DDRMC_CR24)	32	R/W	0000_0000h	34.5.25/ 1421
400A_E064	Control Register 25 (DDRMC_CR25)	32	R/W	0000_0000h	34.5.26/ 1422

Table continues on the next page...

DDRMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E068	Control Register 26 (DDRMC_CR26)	32	R/W	0000_0000h	34.5.27/ 1423
400A_E06C	Control Register 27 (DDRMC_CR27)	32	R (reads 0)	0000_0000h	34.5.28/ 1424
400A_E070	Control Register 28 (DDRMC_CR28)	32	R/W	0000_0000h	34.5.29/ 1424
400A_E074	Control Register 29 (DDRMC_CR29)	32	R/W	0000_0000h	34.5.30/ 1424
400A_E078	Control Register 30 (DDRMC_CR30)	32	R/W	0000_0000h	34.5.31/ 1425
400A_E07C	Control Register 31 (DDRMC_CR31)	32	R/W	0000_0000h	34.5.32/ 1425
400A_E080	Control Register 32 (DDRMC_CR32)	32	R (reads 0)	0000_0000h	34.5.33/ 1426
400A_E084	Control Register 33 (DDRMC_CR33)	32	R/W	0000_0000h	34.5.34/ 1426
400A_E088	Control Register 34 (DDRMC_CR34)	32	R/W	0000_0000h	34.5.35/ 1427
400A_E08C	Control Register 35 (DDRMC_CR35)	32	R/W	0020_0000h	34.5.36/ 1428
400A_E090	Control register 36 (DDRMC_CR36)	32	R/W	0000_0000h	34.5.37/ 1430
400A_E094	Control Register 37 (DDRMC_CR37)	32	R/W	0000_0000h	34.5.38/ 1431
400A_E098	Control Register 38 (DDRMC_CR38)	32	R/W	0000_0000h	34.5.39/ 1433
400A_E09C	Control Register 39 (DDRMC_CR39)	32	R/W	0000_0000h	34.5.40/ 1434
400A_E0A0	Control Register 40 (DDRMC_CR40)	32	R (reads 0)	0000_0000h	34.5.41/ 1435
400A_E0A4	Control Register 41 (DDRMC_CR41)	32	R/W	0000_0000h	34.5.42/ 1436
400A_E0A8	Control Register 42 (DDRMC_CR42)	32	R (reads 0)	0000_0000h	34.5.43/ 1436
400A_E0AC	Control Register 43 (DDRMC_CR43)	32	R (reads 0)	0000_0000h	34.5.44/ 1437
400A_E0B0	Control Register 44 (DDRMC_CR44)	32	R (reads 0)	0000_0000h	34.5.45/ 1437
400A_E0B4	Control Register 45 (DDRMC_CR45)	32	R/W	0000_0000h	34.5.46/ 1438
400A_E0B8	Control Register 46 (DDRMC_CR46)	32	R/W	0000_0000h	34.5.47/ 1438
400A_E0BC	Control Register 47 (DDRMC_CR47)	32	R/W	0000_0000h	34.5.48/ 1439

Table continues on the next page...

DDRMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E0C0	Control Register 48 (DDRMC_CR48)	32	R/W	0000_0000h	34.5.49/ 1440
400A_E0C4	Control Register 49 (DDRMC_CR49)	32	R/W	0000_0000h	34.5.50/ 1440
400A_E0C8	Control Register 50 (DDRMC_CR50)	32	R (reads 0)	0000_0000h	34.5.51/ 1440
400A_E0CC	Control Register 51 (DDRMC_CR51)	32	R/W	0000_0000h	34.5.52/ 1441
400A_E0D0	Control Register 52 (DDRMC_CR52)	32	R/W	0000_0000h	34.5.53/ 1441
400A_E0D4	Control Register 53 (DDRMC_CR53)	32	R (reads 0)	0000_0000h	34.5.54/ 1442
400A_E0D8	Control Register 54 (DDRMC_CR54)	32	R (reads 0)	0000_0000h	34.5.55/ 1442
400A_E0DC	Control Register 55 (DDRMC_CR55)	32	R (reads 0)	0000_0000h	34.5.56/ 1443
400A_E0E0	Control Register 56 (DDRMC_CR56)	32	R (reads 0)	0000_0000h	34.5.57/ 1443
400A_E0E4	Control Register 57 (DDRMC_CR57)	32	R/W	0000_0000h	34.5.58/ 1444
400A_E0E8	Control Register 58 (DDRMC_CR58)	32	R/W	0000_0000h	34.5.59/ 1445
400A_E0EC	Control Register 59 (DDRMC_CR59)	32	R/W	0000_0000h	34.5.60/ 1446
400A_E0F0	Control Register 60 (DDRMC_CR60)	32	R/W	0000_0000h	34.5.61/ 1446
400A_E0F4	Control Register 61 (DDRMC_CR61)	32	R	0000_0000h	34.5.62/ 1447
400A_E0F8	Control Register 62 (DDRMC_CR62)	32	R	0000_0000h	34.5.63/ 1447
400A_E0FC	Control Register 63 (DDRMC_CR63)	32	R	0000_0000h	34.5.64/ 1448
400A_E100	Control Register 64 (DDRMC_CR64)	32	R	0000_0000h	34.5.65/ 1448
400A_E104	Control Register 65 (DDRMC_CR65)	32	R	0000_0000h	34.5.66/ 1448
400A_E108	Control Register 66 (DDRMC_CR66)	32	R/W	0000_0000h	34.5.67/ 1449
400A_E10C	Control Register 67 (DDRMC_CR67)	32	R/W	0000_0000h	34.5.68/ 1449
400A_E110	Control Register 68 (DDRMC_CR68)	32	R (reads 0)	0000_0000h	34.5.69/ 1450
400A_E114	Control Register 69 (DDRMC_CR69)	32	R/W	0000_0000h	34.5.70/ 1450

Table continues on the next page...

DDRMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E118	Control Register 70 (DDRMC_CR70)	32	R/W	0000_0000h	34.5.71/ 1451
400A_E11C	Control Register 71 (DDRMC_CR71)	32	R/W	0000_0000h	34.5.72/ 1452
400A_E120	Control Register 72 (DDRMC_CR72)	32	R/W	0000_0000h	34.5.73/ 1453
400A_E124	Control Register 73 (DDRMC_CR73)	32	R/W	0A00_0000h	34.5.74/ 1454
400A_E128	Control Register 74 (DDRMC_CR74)	32	R/W	0000_0000h	34.5.75/ 1455
400A_E12C	Control Register 75 (DDRMC_CR75)	32	R/W	0000_0000h	34.5.76/ 1456
400A_E130	Control Register 76 (DDRMC_CR76)	32	R/W	0000_0000h	34.5.77/ 1457
400A_E134	Control Register 77 (DDRMC_CR77)	32	R/W	0000_0000h	34.5.78/ 1459
400A_E138	Control Register 78 (DDRMC_CR78)	32	R/W	0000_0000h	34.5.79/ 1460
400A_E13C	Control Register 79 (DDRMC_CR79)	32	R/W	0000_0000h	34.5.80/ 1462
400A_E140	Control Register 80 (DDRMC_CR80)	32	R	0000_0000h	34.5.81/ 1463
400A_E144	Control Register 81 (DDRMC_CR81)	32	W	0000_0000h	34.5.82/ 1464
400A_E148	Control Register 82 (DDRMC_CR82)	32	R/W	0000_0000h	34.5.83/ 1465
400A_E14C	Control Register 83 (DDRMC_CR83)	32	R	0000_0000h	34.5.84/ 1465
400A_E150	Control Register 84 (DDRMC_CR84)	32	R	0000_0000h	34.5.85/ 1466
400A_E154	Control Register 85 (DDRMC_CR85)	32	R	0000_0000h	34.5.86/ 1467
400A_E158	Control Register 86 (DDRMC_CR86)	32	R	0000_0000h	34.5.87/ 1467
400A_E15C	Control Register 87 (DDRMC_CR87)	32	R/W	0000_0000h	34.5.88/ 1468
400A_E160	Control Register 88 (DDRMC_CR88)	32	R/W	0000_0000h	34.5.89/ 1469
400A_E164	Control Register 89 (DDRMC_CR89)	32	R/W	0000_0000h	34.5.90/ 1469
400A_E168	Control Register 90 (DDRMC_CR90)	32	R (reads 0)	0000_0000h	34.5.91/ 1470
400A_E16C	Control Register 91 (DDRMC_CR91)	32	R/W	0000_0000h	34.5.92/ 1470

Table continues on the next page...

DDRMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E170	Control Register 92 (DDRMC_CR92)	32	R/W	0000_0000h	34.5.93/ 1471
400A_E174	Control Register 93 (DDRMC_CR93)	32	R/W	0000_0000h	34.5.94/ 1472
400A_E178	Control Register 94 (DDRMC_CR94)	32	R/W	0000_0000h	34.5.95/ 1474
400A_E17C	Control Register 95 (DDRMC_CR95)	32	R/W	0000_0000h	34.5.96/ 1476
400A_E180	Control Register 96 (DDRMC_CR96)	32	R/W	0000_0000h	34.5.97/ 1477
400A_E184	Control Register 97 (DDRMC_CR97)	32	R/W	0000_0000h	34.5.98/ 1478
400A_E188	Control Register 98 (DDRMC_CR98)	32	R/W	0000_0000h	34.5.99/ 1479
400A_E18C	Control Register 99 (DDRMC_CR99)	32	R/W	0000_0000h	34.5.100/ 1480
400A_E190	Control Register 100 (DDRMC_CR100)	32	R/W	0000_0000h	34.5.101/ 1480
400A_E194	Control Register 101 (DDRMC_CR101)	32	R/W	0000_0000h	34.5.102/ 1481
400A_E198	Control Register 102 (DDRMC_CR102)	32	R/W	0000_0000h	34.5.103/ 1482
400A_E19C	Control Register 103 (DDRMC_CR103)	32	R	0000_0000h	34.5.104/ 1483
400A_E1A0	Control Register 104 (DDRMC_CR104)	32	R/W	0000_0000h	34.5.105/ 1484
400A_E1A4	Control Register 105 (DDRMC_CR105)	32	R/W	0000_0000h	34.5.106/ 1484
400A_E1A8	Control Register 106 (DDRMC_CR106)	32	R/W	0000_0000h	34.5.107/ 1485
400A_E1AC	Control Register 107 (DDRMC_CR107)	32	R/W	0000_0000h	34.5.108/ 1486
400A_E1B0	Control Register 108 (DDRMC_CR108)	32	R	0000_0000h	34.5.109/ 1486
400A_E1B4	Control Register 109 (DDRMC_CR109)	32	R/W	0000_0000h	34.5.110/ 1487
400A_E1B8	Control Register 110 (DDRMC_CR110)	32	R/W	0000_0000h	34.5.111/ 1487
400A_E1BC	Control Register 111 (DDRMC_CR111)	32	R (reads 0)	0000_0000h	34.5.112/ 1488
400A_E1C0	Control Register 112 (DDRMC_CR112)	32	R	0000_0000h	34.5.113/ 1488
400A_E1C4	Control Register 113 (DDRMC_CR113)	32	R/W	0000_0000h	34.5.114/ 1489

Table continues on the next page...

DDRMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E1C8	Control Register 114 (DDRMC_CR114)	32	R/W	0000_0000h	34.5.115/ 1490
400A_E1CC	Control Register 115 (DDRMC_CR115)	32	R/W	0000_0000h	34.5.116/ 1490
400A_E1D0	Control Register 116 (DDRMC_CR116)	32	R (reads 0)	0000_0000h	34.5.117/ 1491
400A_E1D4	Control Register 117 (DDRMC_CR117)	32	R/W	0000_0000h	34.5.118/ 1491
400A_E1D8	Control Register 118 (DDRMC_CR118)	32	R/W	0000_0000h	34.5.119/ 1492
400A_E1DC	Control Register 119 (DDRMC_CR119)	32	R/W	0000_0000h	34.5.120/ 1493
400A_E1E0	Control Register 120 (DDRMC_CR120)	32	R/W	0000_0000h	34.5.121/ 1494
400A_E1E4	Control Register 121 (DDRMC_CR121)	32	R/W	0000_0000h	34.5.122/ 1495
400A_E1E8	Control Register 122 (DDRMC_CR122)	32	R/W	0000_0000h	34.5.123/ 1496
400A_E1EC	Control Register 123 (DDRMC_CR123)	32	R/W	0000_0000h	34.5.124/ 1496
400A_E1F0	Control Register 124 (DDRMC_CR124)	32	R/W	0000_0000h	34.5.125/ 1498
400A_E1F4	Control Register 125 (DDRMC_CR125)	32	R/W	0000_0000h	34.5.126/ 1499
400A_E1F8	Control Register 126 (DDRMC_CR126)	32	R/W	0000_0000h	34.5.127/ 1499
400A_E1FC	Controla Register 127 (DDRMC_CR127)	32	R/W	0000_0000h	34.5.128/ 1500
400A_E200	Control Register 128 (DDRMC_CR128)	32	R (reads 0)	0000_0000h	34.5.129/ 1501
400A_E204	Control Register 129 (DDRMC_CR129)	32	R (reads 0)	0000_0000h	34.5.130/ 1501
400A_E208	Control Register 130 (DDRMC_CR130)	32	R (reads 0)	0000_0000h	34.5.131/ 1501
400A_E20C	Control Register 131 (DDRMC_CR131)	32	R/W	0000_0000h	34.5.132/ 1502
400A_E210	Control Register 132 (DDRMC_CR132)	32	R/W	0000_0000h	34.5.133/ 1502
400A_E214	Control Register 133 (DDRMC_CR133)	32	R (reads 0)	0000_0000h	34.5.134/ 1503
400A_E218	Control Register 134 (DDRMC_CR134)	32	R (reads 0)	0000_0000h	34.5.135/ 1503
400A_E21C	Control Register 135 (DDRMC_CR135)	32	R (reads 0)	0000_0000h	34.5.136/ 1504

Table continues on the next page...

DDRMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E220	Control Register 136 (DDRMC_CR136)	32	R (reads 0)	0000_0000h	34.5.137/ 1504
400A_E224	Control Register 137 (DDRMC_CR137)	32	R/W	0000_0000h	34.5.138/ 1504
400A_E228	Control Register 138 (DDRMC_CR138)	32	R/W	0000_0000h	34.5.139/ 1505
400A_E22C	Control Register 139 (DDRMC_CR139)	32	R/W	0000_0000h	34.5.140/ 1506
400A_E230	Control Register 140 (DDRMC_CR140)	32	R/W	0000_0000h	34.5.141/ 1506
400A_E234	Control Register 141 (DDRMC_CR141)	32	R/W	0000_0000h	34.5.142/ 1507
400A_E238	Control Register 142 (DDRMC_CR142)	32	R/W	0000_0000h	34.5.143/ 1507
400A_E23C	Control Register 143 (DDRMC_CR143)	32	R/W	0000_0000h	34.5.144/ 1507
400A_E240	Control Register 144 (DDRMC_CR144)	32	R/W	0000_0000h	34.5.145/ 1508
400A_E244	Control Register 145 (DDRMC_CR145)	32	R/W	0000_0000h	34.5.146/ 1508
400A_E248	Control Register 146 (DDRMC_CR146)	32	R/W	0000_0000h	34.5.147/ 1509
400A_E24C	Control Register 147 (DDRMC_CR147)	32	R/W	0000_0000h	34.5.148/ 1509
400A_E250	Control Register 148 (DDRMC_CR148)	32	R/W	0000_0000h	34.5.149/ 1510
400A_E254	Control Register 149 (DDRMC_CR149)	32	R/W	0000_0000h	34.5.150/ 1511
400A_E258	Control Register 150 (DDRMC_CR150)	32	R/W	0000_0000h	34.5.151/ 1511
400A_E25C	Control Register 151 (DDRMC_CR151)	32	R/W	0000_0000h	34.5.152/ 1512
400A_E260	Control Register 152 (DDRMC_CR152)	32	R/W	0000_0000h	34.5.153/ 1512
400A_E264	Control Register 153 (DDRMC_CR153)	32	R/W	0000_0000h	34.5.154/ 1513
400A_E268	Control Register 154 (DDRMC_CR154)	32	R/W	0000_0000h	34.5.155/ 1514
400A_E26C	Control Register 155 (DDRMC_CR155)	32	R/W	0000_0000h	34.5.156/ 1515
400A_E270	Control Register 156 (DDRMC_CR156)	32	R	0000_0000h	34.5.157/ 1517
400A_E274	Control Register 157 (DDRMC_CR157)	32	R	0000_0000h	34.5.158/ 1518

Table continues on the next page...

DDRMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E278	Control Register 158 (DDRMC_CR158)	32	R/W	0000_0000h	34.5.159/ 1519
400A_E27C	Control Register 159 (DDRMC_CR159)	32	R/W	0000_0000h	34.5.160/ 1519
400A_E280	Control Register 160 (DDRMC_CR160)	32	R (reads 0)	0000_0000h	34.5.161/ 1520
400A_E284	Control Register 161 (DDRMC_CR161)	32	R (reads 0)	0000_0000h	34.5.162/ 1520
400A_E400	PHY Register 00 (DDRMC_PHY00)	32	R/W	0000_0000h	34.5.163/ 1521
400A_E404	PHY Register 01 (DDRMC_PHY01)	32	R/W	0000_0000h	34.5.164/ 1522
400A_E408	PHY Register 02 (DDRMC_PHY02)	32	R/W	0000_0000h	34.5.165/ 1523
400A_E40C	PHY Register 03 (DDRMC_PHY03)	32	R/W	0000_0000h	34.5.166/ 1524
400A_E410	PHY Register 04 (DDRMC_PHY04)	32	R/W	0000_0000h	34.5.167/ 1524
400A_E428	PHY Register 10 (DDRMC_PHY10)	32	R/W	0000_0000h	34.5.168/ 1525
400A_E42C	PHY Register 11 (DDRMC_PHY11)	32	R/W	0000_0000h	34.5.169/ 1526
400A_E430	PHY Register 12 (DDRMC_PHY12)	32	R/W	0000_0000h	34.5.170/ 1527
400A_E434	PHY Register 13 (DDRMC_PHY13)	32	R	0000_0000h	34.5.171/ 1528
400A_E440	PHY Register 01 (DDRMC_PHY16)	32	R/W	0000_0000h	34.5.172/ 1528
400A_E444	PHY Register 17 (DDRMC_PHY17)	32	R/W	0000_0000h	34.5.173/ 1529
400A_E448	PHY Register 18 (DDRMC_PHY18)	32	R/W	0000_0000h	34.5.174/ 1530
400A_E44C	PHY Register 19 (DDRMC_PHY19)	32	R/W	0000_0000h	34.5.175/ 1531
400A_E450	PHY Register 20 (DDRMC_PHY20)	32	R/W	0000_0000h	34.5.176/ 1531
400A_E468	PHY Register 26 (DDRMC_PHY26)	32	R/W	0000_0000h	34.5.177/ 1532
400A_E46C	PHY Register 27 (DDRMC_PHY27)	32	R/W	0000_0000h	34.5.178/ 1533
400A_E470	PHY Register 28 (DDRMC_PHY28)	32	R/W	0000_0000h	34.5.179/ 1534
400A_E474	PHY Register 29 (DDRMC_PHY29)	32	R	0000_0000h	34.5.180/ 1535

Table continues on the next page...

DDRMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_E480	PHY Register 32 (DDRMC_PHY32)	32	R/W	0000_0000h	34.5.181/ 1535
400A_E484	PHY Register 33 (DDRMC_PHY33)	32	R/W	0000_0000h	34.5.182/ 1536
400A_E488	PHY Register 34 (DDRMC_PHY34)	32	R/W	0000_0000h	34.5.183/ 1537
400A_E48C	PHY Register 35 (DDRMC_PHY35)	32	R/W	0000_0000h	34.5.184/ 1538
400A_E490	PHY Register 36 (DDRMC_PHY36)	32	R/W	0000_0000h	34.5.185/ 1538
400A_E4A8	PHY Register 42 (DDRMC_PHY42)	32	R/W	0000_0000h	34.5.186/ 1539
400A_E4AC	PHY Register 43 (DDRMC_PHY43)	32	R/W	0000_0000h	34.5.187/ 1540
400A_E4B0	PHY Register 44 (DDRMC_PHY44)	32	R/W	0000_0000h	34.5.188/ 1541
400A_E4B4	PHY Register 45 (DDRMC_PHY45)	32	R	0000_0000h	34.5.189/ 1542
400A_E4C4	PHY Register 49 (DDRMC_PHY49)	32	R/W	0000_0000h	34.5.190/ 1542
400A_E4C8	PHY Register 50 (DDRMC_PHY50)	32	R/W	0000_0000h	34.5.191/ 1543
400A_E4D0	PHY Register 52 (DDRMC_PHY52)	32	R/W	0000_0000h	34.5.192/ 1544

34.5.1 Control Register 0 (DDRMC_CR00)

Address: 400A_E000h base + 0h offset = 400A_E000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VERSION															
W																
Reset	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DRAM_CLASS				0							START
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR00 field descriptions

Field	Description
31–16 VERSION	Holds the controller version number.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 DRAM_CLASS	Defines the mode of operation of the controller. Only the following settings are applicable. All other settings are reserved. 0101 LPDDR2 (supports burst lengths of 4 or 8) 0110 DDR3 (supports burst length of 8 only)
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 START	Initiate command processing in the controller. Set to 1 to initiate.

34.5.2 Control Register 1 (DDRMC_CR01)

Address: 400A_E000h base + 4h offset = 400A_E004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															MAX_CS_REG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				MAX_COL_REG				0			MAX_ROW_REG				
W																
Reset	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0

DDRMC_CR01 field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 MAX_CS_REG	Displays the maximum number of chip selects configured for this memory controller
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR01 field descriptions (continued)

Field	Description
11–8 MAX_COL_REG	Defines the maximum width of column address in the DRAM memories. This value can be used to set the col_diff parameter. This parameter is read-only. col_diff = max_col_reg -(number of column bits in memory)
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 MAX_ROW_REG	Defines the maximum width of the memory address bus for the memory controller. **NOTE:** For LPDDR2 memory systems (dram_class = 'b0101), the parameter row_diff holds the difference between the maximum number of row pins configured and the actual number of row pins being. This value can be used to set the row_diff parameter. row_diff = max_row_reg - (number of row bits in memory).

34.5.3 Control Register 2 (DDRMC_CR02)

Address: 400A_E000h base + 8h offset = 400A_E008h

Bit	31–24	23–0
R	0	TINIT
W		
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR02 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–0 TINIT	DRAM TINIT value in cycles.

34.5.4 Control Register 3 (DDRMC_CR03)

Address: 400A_E000h base + Ch offset = 400A_E00Ch

Bit	31–24	23–0
R	0	TINIT3
W		
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR03 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–0 TINIT3	DRAM TINIT3 value in cycles.

34.5.5 Control Register 4 (DDRMC_CR04)

Address: 400A_E000h base + 10h offset = 400A_E010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TINIT4																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR04 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–0 TINIT4	DRAM TINIT4 value in cycles

34.5.6 Control Register 5 (DDRMC_CR05)

Address: 400A_E000h base + 14h offset = 400A_E014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TINIT5																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR05 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–0 TINIT5	DRAM TINIT5 value in cycles.

34.5.7 Control Register 6 (DDRMC_CR06)

Address: 400A_E000h base + 18h offset = 400A_E018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR06 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.8 Control Register 7 (DDRMC_CR07)

Address: 400A_E000h base + 1Ch offset = 400A_E01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR07 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.9 Control Register 8 (DDRMC_CR08)

Address: 400A_E000h base + 20h offset = 400A_E020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR08 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.10 Control Register 9 (DDRMC_CR09)

Address: 400A_E000h base + 24h offset = 400A_E024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							NO_MRR	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR09 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 NO_MRR	Disables MRR commands until DLL initialization is complete. This parameter does not affect the reading of any other MRs such as MR0 or MR4. This parameter is only applicable when the MC is programmed for the following memory systems: LPDDR2 (dram_class = ’b0101) 0 No restrictions on MRR commands during initialization. 1 Do not issue MRR commands during DLL initialization of the DRAM memories.
23–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.11 Control Register 10 (DDRMC_CR10)

Address: 400A_E000h base + 28h offset = 400A_E028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	TRST_PWRON																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR10 field descriptions

Field	Description
31–0 TRST_PWRON	Defines the number of memory clocks that memory will be held in reset during the power-on initialization sequence. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)

34.5.12 Control Register 11 (DDRMC_CR11)

Address: 400A_E000h base + 2Ch offset = 400A_E02Ch

Bit	31–0
R/W	CKE_INACTIVE
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR11 field descriptions

Field	Description
31–0 CKE_INACTIVE	Defines the number of memory clocks after reset before the CKE signal will be active. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class =b0110) **NOTE:** For the high-speed controller architecture, the observed minimum values may be one controller clock greater than the programmed value due to the nature of the half rate clocking

34.5.13 Control Register 12 (DDRMC_CR12)

Address: 400A_E000h base + 30h offset = 400A_E030h

Bit	31–16	15–13	12–8	7–6	5–0
R	0	0	WRLAT	0	CASLAT_LIN
W					
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 0	0 0	0 0 0 0 0 0

DDRMC_CR12 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 WRLAT	Defines the write latency from when the write command is issued to the time the write data is presented to the DRAM memories, in controller clocks.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 CASLAT_LIN	Sets latency from read command send to data receive from/to controller. Bit (0) is half-cycle increment and the upper bits define memory CAS latency for the controller.

34.5.14 Control Register 13 (DDRMC_CR13)

Address: 400A_E000h base + 34h offset = 400A_E034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	TRC								TRRD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TCCD					0					TBST_INT_INTERVAL		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR13 field descriptions

Field	Description
31–24 TRC	Defines the DRAM period between active commands for the same bank, in memory clocks.
23–16 TRRD	Defines the DRAM activate to activate delay to different bank groups, in memory clocks.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 TCCD	Defines the minimum delay between CAS commands, in memory clocks. This value is loaded into a counter when a memory burst is issued and a new command may be issued when the counter reaches 0. **NOTE:** Since the counter is initially loaded with the value “tccd-1”, if this parameter is cleared to ’b0, the counter will underflow on load. This will result in setting the counter to its maximum value. Therefore, the user should set this parameter to ’b1 as a minimum.
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 TBST_INT_ INTERVAL	DRAM burst interrupt interval value in cycles.

34.5.15 Control Register 14 (DDRMC_CR14)

Address: 400A_E000h base + 38h offset = 400A_E038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		TFAW						0			TRP					0				TWTR				TRAS_MIN							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR14 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR14 field descriptions (continued)

Field	Description
29–24 TFAW	Defines the DRAM tFAW term, in memory clocks
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 TRP	Defines the DRAM pre-charge / preactivate command time, in memory clocks.If the memory specification defines both a single-bank and an all-banks pre-charge value, this parameter should be programmed to the single-bank value and the trp_ab parameter should be programmed to the all-banks value.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 TWTR	Sets the number of memory clocks needed to switch from a write to a read operation.
7–0 TRAS_MIN	Defines the DRAM minimum row activate time, in memory clocks

34.5.16 Control Register 15 (DDRMC_CR15)

Address: 400A_E000h base + 3Ch offset = 400A_E03Ch

Bit	31–16	15–0
R	0	0
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR15 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.17 Control Register 16 (DDRMC_CR16)

Address: 400A_E000h base + 40h offset = 400A_E040h

Bit	31–29	28–24	23–20	19–16	15–0
R	0	TMRD	0	TRTP	0
W					
Reset	0 0 0	0 0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR16 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 TMRD	Defines the minimum number of memory clocks required between two mode register write commands. This is the time required to complete the write operation to the mode register. There are two parameters that control the timing after a mode register write: tmod and tmrd. This parameter (tmrd) is typically the shorter of the two timing delays. The MC will wait this delay after a mode register command before sending the next mode register command. The MC does not need to wait for the longer tmod timer to expire since termination is not used for mode register commands. **NOTE:** For LPDDR2 memory systems (dram_class = 'b0101), this parameter should be programmed to the tMRW value from the memory specification
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 TRTP	The meaning of this parameter depends on the memory system being used: • For LPDDR2 memories: Program to the tRTP (read to pre-charge time) value from the memory specification. • For DDR3 memories: Program to the tRTP (read to pre-charge time) value from the memory specification
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.18 Control Register 17 (DDRMC_CR17)

Address: 400A_E000h base + 44h offset = 400A_E044h

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="7">0</td><td colspan="17" rowspan="2">TRAS_MAX</td><td colspan="8" rowspan="2">TMOD</td></tr>
<tr><td>W</td><td colspan="7"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

DDRMC_CR17 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24–8 TRAS_MAX	Defines the DRAM maximum row active time, in memory clocks.The user should set this value to the timing parameter in the memory specification. In practical use, the Memory Controller will use the value in this parameter with adjustments made for write recovery time, memory burst length and any internal delays if required
7–0 TMOD	Defines the number of memory clocks of delay after a mode register write to any non-mode register write command. For write leveling, this is defined as the number of memory clocks of wait time after a MRS command to the ODT enable. There are two parameters that control the timing after a mode register write: tmod and tmrd. This parameter (tmod) is typically the longer timing delay (defined by the memory specification) and controls the spacing of a mode register command to any other memory command. The MC will wait this delay after a mode register command before sending any other command to memory since the controller has no way of knowing whether or not termination (ODT) will be used. **NOTE:** There are two parameters that control the timing after a mode register write: tmod and tmrd. This parameter (tmod) is typically the longer timing delay (defined by the memory specification) and controls the spacing of a mode register command to any other memory command. The MC will wait this delay after a mode register command before sending any other command to memory since the controller has no way of knowing whether or not termination (ODT) will be used.

34.5.19 Control Register 18 (DDRMC_CR18)

Address: 400A_E000h base + 48h offset = 400A_E048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0			TCKESR					0					TCKE		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR18 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 TCKESR	Defines the minimum number of cycles that CKE must be held low during selfrefresh, in memory clocks. If the memory specification does not define a tCKESR, then the tckesr parameter should be programmed to the tCKE value from the memory specification
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 TCKE	Defines the minimum CKE pulse width, in memory clocks

34.5.20 Control Register 19 (DDRMC_CR19)

Address: 400A_E000h base + 4Ch offset = 400A_E04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR19 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.21 Control Register 20 (DDRMC_CR20)

Address: 400A_E000h base + 50h offset = 400A_E050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							AP	0							WRITEINTERP
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR20 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 AP	Enables auto pre-charge mode for DRAM memories **NOTE:** This parameter may not be modified after the start parameter has been asserted.When using auto-precharge per command,this parameter must be cleared to ’b0. 0 Disable auto pre-charge mode. Memory banks will stay open until another request requires this bank, the maximum open memory clocks has elapsed,or a refresh command closes all the blanks. 1 Enable auto pre-charge mode. All read and write transactions must be terminated by an auto pre-charge command. If a transaction consists of multiple read or write bursts, only the last command is issued with an auto pre-charge.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 WRITEINTERP	Defines whether the memory controller can interrupt a write burst with a read command. Some memories do not allow this functionality. • For LPDDR2 memories: This parameter must be cleared to ’b0. • For DDR3 memories: This parameter must be cleared to ’b0. For all memory types: 0 The memory does not support read commands interrupting write commands. 1 The memory does support read commands interrupting write commands
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.22 Control Register 21 (DDRMC_CR21)

Address: 400A_E000h base + 54h offset = 400A_E054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								TRCD_INT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							TRAS_ LOCKOUT	0							CCMAP
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR21 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 TRCD_INT	Defines the DRAM RAS to CAS delay, in memory clocks.
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 TRAS_ LOCKOUT	Defines the tRAS lockout setting for the DRAM memory, in memory clocks. tRAS lockout allows the memory controller to execute auto pre-charge commands before the tras_min_X parameter has expired. Note: This parameter may only be applicable to DDR1 memory systems (dram_class = 0000). 0 tRAS lockout not supported by memory. 1 tRAS lockout supported by memory.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 CCMAP	Enables concurrent auto pre-charge. Some DRAM memories do not allow one bank to be auto pre-charged while another bank is reading or writing. The JEDEC standard allows concurrent auto precharge. The user should set this parameter if the DRAM memory supports this feature. Note: This parameter can only be set if the memory supports concurrent AP for all transactions. Some memories only support concurrent AP for writes. In this case, these memories require that reads to the same CS cannot be issued before the tWR delay has expired. If the memories being used do not support CCMAP for all transactions, CCMAP must be cleared to 0, disabling CCMAP for all transfers. 0 Disable concurrent auto precharge. 1 Enable concurrent auto precharge.

34.5.23 Control Register 22 (DDRMC_CR22)

Address: 400A_E000h base + 58h offset = 400A_E058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										TDAL						0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR22 field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 TDAL	Defines the auto pre-charge write recovery when auto pre-charge is enabled, in memory clocks. This is defined internally as tRP (pre-charge time) + auto pre-charge write recovery time.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.24 Control register 23 (DDRMC_CR23)

Address: 400A_E000h base + 5Ch offset = 400A_E05Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					BSTLEN			0				TMRR				TDLL															
W																																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR23 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 BSTLEN	Defines the memory burst length encoding that will be programmed into the DRAM memories at initialization. The mode is programmed in the dram_class parameter. The reset value of this bit can change after reset based on internal logic. All settings except 010 and 011 are reserved. Note: For this memory controller, the user is advised to run in 8 word mode for best performance. Note: This parameter will be cleared to 0x0 on reset, but will change to the default value on the first controller clock edge after reset. Note: The reset value of this bit can change after reset based on internal logic. 010 4 memory words. Applicable for these memory systems: LPDDR2 (dram_class = 0101). 011 8 memory words. Applicable for these memory systems: LPDDR2 (dram_class = 0101) DDR3 (dram_class = 0110)
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 TMRR	Defines the DRAM tMRR term, in memory clocks. This parameter is only applicable when the MC is programmed for the following memory systems: LPDDR2 (dram_class = 0101)

Table continues on the next page...

DDRMC_CR23 field descriptions (continued)

Field	Description
15–0 TDLL	Defines the DRAM DLL lock delay, in memory clocks. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110). Note: For DDR3 memory systems (dram_class = 0110), this parameter is only applicable during a frequency scaling procedure where is it defined as 512 clocks.

34.5.25 Control Register 24 (DDRMC_CR24)

Address: 400A_E000h base + 60h offset = 400A_E060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ADD_MIR	0							DIMM_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											TRP_AB				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR24 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 ADD_MIR	Indicates which chip selects support address mirroring. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110) 0 Standard pinout (for each chip select). 1 Mirrored wiring (for each chip select).
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 DIMM_EN	Enables registered DIMM operations to control the address and command pipeline of the controller. **NOTE:** For this controller, when the MC is programmed for DDR3 memory systems (dram_class = 0110), enabling this bit is not sufficient for registered DIMM support. 0 Normal operation 1 Enable registered DIMM operation
15–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 TRP_AB	Defines the DRAM TRP time for all banks, in memory clocks. The memory specification defines both a single-bank and an all-banks pre-charge value, this parameter should be programmed to the all-banks value and the trp parameter should be programmed to the single-bank value. If the memory specification

Table continues on the next page...

DDRMC_CR24 field descriptions (continued)

Field	Description
	does NOT specify an all-banks pre-charge value, both parameters should be programmed to the same value.

34.5.26 Control Register 25 (DDRMC_CR25)

Address: 400A_E000h base + 64h offset = 400A_E064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															TREF_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							AUTO_RFM	0							
W																AREF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR25 field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 TREF_EN	Issue auto-refresh commands to the DRAMs at the interval defined in the TREF parameter. Set to 1 to Enables refresh commands. If command refresh mode is configured,then refresh commands will be automatically issued based on the tref parameter value and any refresh commands sent through the command interface or the register interface. Refreshes will still occur even if the DRAM memories have been placed in power-down state. Note: Even with this parameter cleared to 0, some refresh commands may still be issued to memory. This only controls the automatic refreshes issued every tref memory clocksenable.

Table continues on the next page...

DDRMC_CR25 field descriptions (continued)

Field	Description
	0 Disable refresh commands. 1 Enable refresh commands.
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 AUTO_RFM	Sets the mode for when the automatic refresh will occur. If the auto_refmode parameter is set and a refresh is required to memory, the memory controller will delay this refresh until the end of the current transaction (if the transaction is fully contained inside a single page), or until the current transaction hits the end of the current page. 0 Issue refresh on the next memory burst boundary, even if the current command is not complete. 1 Issue refresh on the next command boundary.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 AREF	Initiates an automatic refresh to the DRAM memories based on the setting of the auto_refmode parameter. If there are any open banks when this parameter is set, the Memory Controller will automatically close these banks before issuing the auto-refresh command. This parameter is write-only and will always read back as 0x0. 0 No action. 1 Issue refresh to the DRAM memories.

34.5.27 Control Register 26 (DDRMC_CR26)

Address: 400A_E000h base + 68h offset = 400A_E068h

Bit	31–16	15–10	9–0
R	TREF	0	TRFC
W	TREF		TRFC
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

DDRMC_CR26 field descriptions

Field	Description
31–16 TREF	Defines the DRAM time between refresh commands, in memory clocks. This parameter sets the average interval between refreshes.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–0 TRFC	Defines the DRAM refresh command time 0, in memory clocks.

34.5.28 Control Register 27 (DDRMC_CR27)

Address: 400A_E000h base + 6Ch offset = 400A_E06Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR27 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.29 Control Register 28 (DDRMC_CR28)

Address: 400A_E000h base + 70h offset = 400A_E070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TREF_INT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR28 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 TREF_INT	Sets the time delay, in memory clocks, between when refresh commands are issued to different chip selects. If tref_int is cleared to 000, then the refresh per chip select logic will be disabled and all chip selects will be refreshed simultaneously. The tref_int parameter value must be less than the trfc parameter value.

34.5.30 Control Register 29 (DDRMC_CR29)

Address: 400A_E000h base + 74h offset = 400A_E074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TPDEX															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR29 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 TPDEX	Defines the DRAM power-down exit command period, in memory clocks. For LPDDR2 memories: Program to the tXP value from the memory specification. For DDR3 memories: There are two parameters to specify power-down exit time, tpdex and txpdll. This parameter should be programmed to the tXP value from the memory specification. The memory controller will use this value for power-down exit time when the parameter mr0_data_X [12] bit (Dll Control for PreCharge) is set to 1 (fast exit).

34.5.31 Control Register 30 (DDRMC_CR30)

Address: 400A_E000h base + 78h offset = 400A_E078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPDLL															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR30 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 TXPDLL	Defines the DRAM power-down exit command period, in memory clocks. There are two parameters to specify power-down exit time, tpdex and txpdll. This parameter should be programmed to the tXPDLL value from the memory specification. The memory controller will use this value for powerdown exit time when the parameter mr0_data_X [12] bit (Dll Control for PreCharge) is cleared to 0 (slow exit). This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110)

34.5.32 Control Register 31 (DDRMC_CR31)

Address: 400A_E000h base + 7Ch offset = 400A_E07Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXSNR																TXSR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR31 field descriptions

Field	Description
31–16 TXSNR	Defines the DRAM time from a selfrefresh exit to a command that does not require the memory DLL to be locked, in memory clocks. Note: For DDR3 memory systems (dram_class = 0110), this parameter should be programmed to the tXS value from the memory specification. And since the DDR3 JEDEC specification defines tXPR = max (tXS, 5*tCK), this parameter can also be used to account for the memory timing value tXPR. The user must set the txsnr parameter to at least 0x5 for this relationship.

Table continues on the next page...

DDRMC_CR31 field descriptions (continued)

Field	Description
15–0 TXSR	Defines the DRAM time from a selfrefresh exit to an active command that requires the memory DLL to be locked, in memory clocks. The meaning of this parameter depends on the memory system being used: • For LPDDR2 memories: Set this parameter and the txsnr parameter to the same value. • For DDR3 memories: Set this parameter to the difference between the tXSDLL and tRCD values in the memory specification.

34.5.33 Control Register 32 (DDRMC_CR32)

Address: 400A_E000h base + 80h offset = 400A_E080h

Bit	31–16	15–0
R	0	0
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR32 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.34 Control Register 33 (DDRMC_CR33)

Address: 400A_E000h base + 84h offset = 400A_E084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					CKE_DELAY			0							EN_QK_SREF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							SREF_EX_NOREF	0							PWUP_SREF_EX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR33 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 CKE_DELAY	Additional cycles to delay CKE for status reporting.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 EN_QK_SREF	When this bit is set to 1, the memory initialization sequence may be interrupted and the memory may enter self-refresh mode. This is used to place the memories into self-refresh mode when a power loss is detected during the initialization process. Setting this parameter does not initiate a self-refresh entry during initialization. However, if a self-refresh entry is requested during initialization, it will only occur if this parameter is also set to 1. 0 Continue memory initialization. 1 Interrupt memory initialization and enter self-refresh mode.
15–9 Reserved	This field is reserved.
8 SREF_EX_ NOREF	When a self-refresh exit command is executed, an automatic refresh is requested. By setting this bit, the automatic refresh request is inhibited. 0 Allow automatic refresh. 1 Inhibit automatic refresh.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 PWUP_SREF_ EX	Allows the controller to exit low power by executing a self-refresh exit instead of the full memory initialization. This parameter provides a means to skip full initialization when the DRAM memories are in a known self-refresh state. 0 Disable. 1 Enable. If this option is being used.

34.5.35 Control Register 34 (DDRMC_CR34)

Address: 400A_E000h base + 88h offset = 400A_E088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												CKSRX			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CKSRE				0						0	LP_REFEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR34 field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 CKSRX	Sets the number of controller clocks to hold the controller clock stable before exiting self-refresh mode.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 CKSRE	Sets the number of controller clocks to hold the controller clock stable after entering self-refresh mode. The controller clock will run for a minimum of cksre clocks after CKE falls.
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LP_REFEN	Sets whether refreshes will occur while the memory controller is in one of the power-down modes. Bit [0] controls CS0. Note that refreshes will not occur while in any of the self-refresh modes. Note: This parameter is active low. 0 Refreshes still occur. 1 Refreshes do not occur.

34.5.36 Control Register 35 (DDRMC_CR35)

Address: 400A_E000h base + 8Ch offset = 400A_E08Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				LP_ARBST				0		LP_ST														Reserved							
W																	LP_CMD															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR35 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

27–24
LP_ARBST
	Low power arbiter state status register, which defines which interface has control of the LPC module. This parameter is readonly.
	• Bit [3] - Software lock indicator
	  • 0 - No lock is in place
	  • 1 - The software programmable interface has locked the arbiter. Only a low power command issued through the register interface without the lock bit set, or a high-priority command on the external pin interface can clear this lock.

23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 LP_ST	Note: If an active power-down was requested, the memory will enter either active or pre-charge power-down depending on the state of the rows. If there are no active rows, pre-charge power-down will be entered. Even if pre-charge powerdown is used, this parameter will reflect the requested active power-down state. Low power state status register. This value is also reflected in the output signal lp_ext_state

Table continues on the next page...

DDRMC_CR35 field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>and is not valid when the memory controller is processing a low power command. This parameter is read-only.
Bit [6] = Valid. This bit will be cleared to 0 when a command is accepted on the software programmable or external pin interfaces, and remain at 0 until the command has completed.
0 = Invalid, low power state currently in transition
1 = Valid, stable low power state
Bits [5:0] = Low power state.
000000 = Idle
000001 = Active Power-Down
000010 = Active Power-Down with Memory Clock Gating
000011 = Pre-Charge Power-Down
000100 = Pre-Charge Power-Down with Memory Clock Gating
000101 = Self-Refresh
000110 = Self-Refresh with Memory Clock Gating
000111 = Self-Refresh with Memory and Controller Clock Gating
100000 = Idle, with lock
100001 = Active Power-Down, with lock
100010 = Active Power-Down with Memory Clock Gating, with lock
100011 = Pre-Charge Power-Down, with lock
100100 = Pre-Charge Power-Down with Memory Clock Gating, with lock
100101 = Self-Refresh, with lock
100110 = Self-Refresh with Memory Clock Gating, with lock
100111 = Self-Refresh with Memory and Controller Clock Gating, with lock
All other settings are Reserved</td></tr>
<tr><td>15–8
LP_CMD</td><td>Defines the low power command requested through the software programmable interface. When this command is completed, the low power command complete interrupt (bit 10) in the INT_STAT parameter will be set to 1. Re-programming attempts to this parameter until the interrupt is set will be ignored. This parameter is write-only and will always read back as 0x0. The bits are defined as follows:
Bit [7] = Lock.
0 = No action
1 = Lock command
Bit [6] = Controller clock gating.
0 = No action
1 = Gate the controller clock
Bit [5] = Memory clock gating.
0 = No action
1 = Gate the memory clock
Bits [4:2] = Low power state.
000 = Active power-down
001 = Pre-charge power-down
010 = Self-refresh
All other settings are Reserved
Bit [1] = Entry command.
0 = No action
1 = Enter the specified state
Bit [0] = Exit command.
0 = No action
1 = Exit the specified state</td></tr>
</table>

Table continues on the next page...

DDRMC_CR35 field descriptions (continued)

Field	Description
7–0 Reserved	This field is reserved.

34.5.37 Control register 36 (DDRMC_CR36)

Address: 400A_E000h base + 90h offset = 400A_E090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														LP_AMEMEN	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					LP_AEXEN			0					LPAUTO		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR36 field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 LP_AMEMEN	Enables memory clock gating when entering each of the low power modes automatically. A low power mode is only entered automatically if the associated bit in the lpauto parameter is also set to 1 when the idle timer expires. Note: This parameter does not apply to the ?Self-Refresh with Memory and Controller Clock Gating,? low power mode. If this timer expires and lpauto [2] is set to 1, the memory clock will always be gated. • Bit [1] = Self-Refresh • Bit [0] = Power-Down Note: The MC will issue entry into active power-down or pre-charge powerdown depending on the state of the rows at the time the command is required. This differentiation is only applicable when the MC is programmed for the following memory systems: LPDDR2 (dram_class = ’b0101). 0 (For all bits) Disable memory clock gating when automatically entering into this low power state. 1 (For all bits) Enable memory clock gating when automatically entering into this low power state.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 LP_AEXEN	Enables automatic exit from each of the low power states when the associated idle timer expires. The low power state will be exited automatically when a read or write command enters the command queue in the memory controller core. Note: Automatic exit will occur regardless of the interface that initiated the low power state entry (external pin interface, software programmable interface or automatic interface). Note: Automatic exit will not interrupt a command sequence such as frequency scaling or ZQ calibration, or a shutdown. • Bit [2] = Self-Refresh with Memory and Controller Clock Gating

Table continues on the next page...

DDRMC_CR36 field descriptions (continued)

Field	Description
	• Bit [1] = Self-Refresh • Bit [0] = Power-Down Note: Note: For power-down, enabling the bit will cause exit from either the active power-down or pre-charge power-down modes. This differentiation is only applicable when the MC is programmed for the following memory systems:LPDDR2 (dram_class = 0101). 0 (For all bits) Disable automatic exit from this low power state. 1 (For all bits) Enable automatic exit from this low power state. The state will be exited automatically when a read or write command enters the memory controller core command queue.
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 LPAUTO	Enables automatic entry into each of the low power states when the associated idle timer expires.For power-down or self-refresh modes, the memory clock will also be gated when the timer expires if the associated bit in the lp_amemen parameter is set to 1. For “Self-Refresh with Memory and Controller Clock Gating,” memory clock gating will always be included. • Bit [2] = Self-Refresh with Memory and Controller Clock Gating • Bit [1] = Self-Refresh • Bit [0] = Power-Down Note: For power-down, the MC will issue entry into active power-down or pre-charge power-down depending on the state of the rows at the time the command is required. This differentiation is only applicable when the MC is programmed for the following memory systems:LPDDR2 (dram_class = 0101). 0 (For all bits) Disable automatic entry into this low power state. 1 (For all bits) Enable automatic entry into this low power state. The state will be entered automatically when the proper counters expire.

34.5.38 Control Register 37 (DDRMC_CR37)

Address: 400A_E000h base + 94h offset = 400A_E094h

Bit	31–24	23–16	15–12	11–0
R	LP_A_SR_MC_IDL	LP_A_SR_IDL	0	LP_A_PD_IDL
W				
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR37 field descriptions

Field	Description
31–24 LP_A_SR_MC_IDL	“Self-Refresh with Memory and Controller Clock Gating” auto entry refresh cycles. This parameter defines the number of refresh cycles that can elapse before the memory controller will automatically issue an entry into the “Self-Refresh with Memory and Controller Clock Gating” low power state. Clearing this parameter to 0x0 disables automatic entry, even if the associated bit in the lpauto parameter is set to 1.
23–16 LP_A_SR_IDL	“Self-Refresh” or “Self-Refresh with Memory Clock Gating” auto entry refresh cycles. This parameter defines the number of refresh cycles that can elapse before the memory controller will automatically issue an entry into the “Self-Refresh” or “Self-Refresh with Memory Clock Gating” low power state. Clearing this parameter to 0x0 disables automatic entry, even if the associated bit in the lp_auto_entry_en parameter is set to 1.

Table continues on the next page...

DDRMC_CR37 field descriptions (continued)

Field	Description
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 LP_A_PD_IDL	Defines the power-down mode auto entry controller clocks. This parameter relates to all of the following states: • “Active Power-Down” • “Active Power-Down with Memory Clock Gating” • “Pre-Charge Power-Down” • “Pre-Charge Power-Down with Memory Clock Gating” The MC will issue an active power-down unless all of the pages are closed. In that case, the MC will issue a pre-charge power-down. This parameter defines the number of idle controller clocks that can elapse before the memory controller will automatically issue an entry into the approprte powerdown low power state. This value will be loaded into the associated counter when a command enters the command queue. Clearing this parameter to 0x0 disables automatic entry, even if the associated bit in the lp_auto_entry_en parameter is set to 1.

34.5.39 Control Register 38 (DDRMC_CR38)

Address: 400A_E000h base + 98h offset = 400A_E098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							FRQ_CHG_HOEN
W								FREQ_CHG_HOCL								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							FREQ_CHG_EN	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR38 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 FREQ_CHG_ HOCL	Handshaking signal from the system to the MC that the system has completed whatever operations were required (such as training) following frequency scaling and is ready for normal operation. This parameter is write-only and will always read back as 0x0. Note: This parameter is only relevant if the freq_change_done_hold_en parameter is set to 1.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 FRQ_CHG_ HOEN	Enables the system to perform any required processes following frequency scaling before the MC resumes normal operation. After the frequency has been changed and the PHY has been re-locked, the frequency change complete interrupt (bit 26) will be set to ’b1 in the int_stat parameter. If this parameter is

Table continues on the next page...

DDRMC_CR38 field descriptions (continued)

Field	Description
	set to 1, the MC will be held off from normal operation until the system sets the freq_change_done_hold_clear parameter. 0 Disable 1 Enable
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 FREQ_CHG_EN	Enables hardware dynamic frequency scaling. 0 Disable 1 Enable
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.40 Control Register 39 (DDRMC_CR39)

Address: 400A_E000h base + 9Ch offset = 400A_E09Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHY_INI_COM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHY_INI_STA								0						FRQ_CH_DLLOFF	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR39 field descriptions

Field	Description
31–16 PHY_INI_COM	Holds the DFI tinit_complete timing parameter, the maximum number of DFI clocks after the de-assertion of the dfi_init_start signal to the re-assertion of the dfi_init_complete signal.
15–8 PHY_INI_STA	Holds the DFI tinit_start timing parameter, the number of DFI clocks from the assertion of the dfi_init_start signal on the DFI until the PHY must respond by de-asserting the dfi_init_complete signal.
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 FRQ_CH_ DLLOFF	Defines if the memory DLL must be on. Set each bit to 1 to require DLL off. • 0 = DLL must be on • 1 = DLL must be off This parameter is only applicable when the MC is programmed for the following memory systems:DDR3 (dram_class = 0110).

34.5.41 Control Register 40 (DDRMC_CR40)

Address: 400A_E000h base + A0h offset = 400A_E0A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							CURR_REG_CP	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR40 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 CURR_REG_CP	Indicates which set of timing parameters are being used by the memory controller.
23–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.42 Control Register 41 (DDRMC_CR41)

Address: 400A_E000h base + A4h offset = 400A_E0A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															PHY_INI_STRT_INI_DIS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR41 field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 PHY_INI_STRT_ INI_DIS	Disables the assertion of the MC frequency change request during DFI initialization. The value of this parameter does not affect the assertion of the frequency change request signal during the frequency scaling operation. The frequency change request signal is optional and only used by MCs that support the frequency change protocol. 0 Allow the frequency change request signal to assert. 1 Disable the frequency change request signal assertion.

34.5.43 Control Register 42 (DDRMC_CR42)

Address: 400A_E000h base + A8h offset = 400A_E0A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR42 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 Reserved	This read-only field is reserved and always has the value 0.

34.5.44 Control Register 43 (DDRMC_CR43)

Address: 400A_E000h base + ACh offset = 400A_E0ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR43 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 Reserved	This read-only field is reserved and always has the value 0.

34.5.45 Control Register 44 (DDRMC_CR44)

Address: 400A_E000h base + B0h offset = 400A_E0B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR44 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 Reserved	This read-only field is reserved and always has the value 0.

34.5.46 Control Register 45 (DDRMC_CR45)

Address: 400A_E000h base + B4h offset = 400A_E0B4h

Bit	31–26	25–0
R	0	WRMD
W		
Reset	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR45 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25–0 WRMD	Issues mode register write(s) to the specified mode register(s) and memories. When the write has been completed, the mode register write complete interrupt (bit 21) will be set to 1 in the int_stat parameter. Any errors will be reported in the MRW_STAT bits. This parameter will be reset to 0x0 on the controller clock after it is changed and can not be reprogrammed until the interrupt occurs. **NOTE:** Software internal logic can also update this bit.

34.5.47 Control Register 46 (DDRMC_CR46)

Address: 400A_E000h base + B8h offset = 400A_E0B8h

Bit	31–25	24–8	7–0
R	0	RMD	MRW_STAT
W			
Reset	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

DDRMC_CR46 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24–8 RMD	Holds the memory mode register 2 data written during memory initialization, during a frequency scaling procedure, or when the WRMD parameter bit [25] is asserted with bit [16] or bit [17] set to 1. The meaning of this parameter depends on the memory system being used: • For LPDDR2 memories: (MR2) • For DDR3 memories: (MR2) **NOTE:** Software internal logic can also update this bit.
7–0 MRW_STAT	Provides status of the write mode register request issued through the assertion of WRMD(write_modereg) bit . This parameter is read only and is valid when the mode register write complete interrupt (bit 21) is set to 1 in the int_stat parameter. • Bits [7:3] = Reserved • Bit [2] = Reserved

Table continues on the next page...

DDRMC_CR46 field descriptions (continued)

Field	Description
	• Bit [1] = A mode register write was requested for the PASR mode registers (MR16, MR17) when the memory controller is not programmed for LPDDR2 memory systems (dram_class is not programmed to 0101). • Bit [0] = WRMD(write_modereg) programming error. This bit will be set if no mode register write type is specified (WRMD bits [23:16]) but the mode register write was triggered (WRMD bit [25] = 1).

34.5.48 Control Register 47 (DDRMC_CR47)

Address: 400A_E000h base + BCh offset = 400A_E0BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	REF_AU_TMPCHK								0				AU_TMPCHK_VAL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PERI_MRR_DA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR47 field descriptions

Field	Description
31–24 REF_AU_ TMPCHK	Sets the maximum number of refresh commands between automatic mode register reads of MR4. Setting this parameter to 0x0 will disable automatic polling of the MR4. On each expiration of this counter, the MR4 for the next chip select in order will be read and stored into the relevant bits of the auto_tempchk_val parameter. This parameter is only applicable when the MC is programmed for the following memory systems: LPDDR2 (dram_class = 0101)
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–16 AU_TMPCHK_ VAL	Holds the data read from MR4 (bits OP7, OP2, OP1 and OP0) for all chip selects. This information is gathered from the LPDDR2 memories automatically through MRR commands. The frequency of updates to this parameter is defined in the refresh_per_auto_tempchk parameter. This parameter is read-only. • Bits [3:0] = MR4 data for CS 0. This parameter is only applicable when the MC is programmed for the following memory systems: • LPDDR2 (dram_class = 0101)
15–0 PERI_MRR_DA	Data and chip returned from the mode register read issued as a result of setting the bit RMD [16] to 1. The contents of this parameter are only valid when the peripheral mode register read complete interrupt (bit 18) in the int_stat parameter is set to 1. This parameter is read-only. • Bits [15:8] = MRR chip information • Bits [7:0] = Mode register data This parameter is only applicable when the MC is programmed for the following memory systems: LPDDR2 (dram_class = 0101)

34.5.49 Control Register 48 (DDRMC_CR48)

Address: 400A_E000h base + C0h offset = 400A_E0C0h

Bit	31–16	15–0
R/W	MR1_DA_0	MR0_DA_0
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR48 field descriptions

Field	Description
31–16 MR1_DA_0	Holds the memory mode register 1 data written during memory initialization, or when the WRMD parameter bit [25] is asserted with bit [16] or bit [17] set to 1.
15–0 MR0_DA_0	Holds the memory mode register 0 data written during memory initialization, or when the WRMD parameter bit [25] is asserted with bit [16] or bit [17] set to 1.

34.5.50 Control Register 49 (DDRMC_CR49)

Address: 400A_E000h base + C4h offset = 400A_E0C4h

Bit	31–16	15–0
R	0	MR2_DA_0
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR49 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 MR2_DA_0	Holds the memory mode register 2 data written during memory initialization,or when the WRMD parameter bit [25] is asserted with bit [16] or bit [17] set to 1. The meaning of this parameter depends on the memory system being used: • For LPDDR2 memories: (MR2) • For DDR3 memories: (MR2)

34.5.51 Control Register 50 (DDRMC_CR50)

Address: 400A_E000h base + C8h offset = 400A_E0C8h

Bit	31–16	15–0
R	0	0
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR50 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 Reserved	This read-only field is reserved and always has the value 0.

34.5.52 Control Register 51 (DDRMC_CR51)

Address: 400A_E000h base + CCh offset = 400A_E0CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MR3_DA0																MR_SINDA0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR51 field descriptions

Field	Description
31–16 MR3_DA0	Holds the memory mode register 3 data for chip select 0 written during memory initialization or when the WRMD parameter bit [25] is asserted with bit [16] or bit [17] set to 1. The meaning of this parameter depends on the memory system being used: • For LPDDR2 memories: (MR3) • For DDR3 memories: (MR3) **NOTE:** Note: When programmed for DDR3 memory systems (dram_class = 0110), bit [2] of MR3 cannot be programmed through this parameter. This bit is controlled by the read leveling state machine.
15–0 MR_SINDA0	Holds the data to be programmed into a single memory mode register for chip select 0. This parameter is used when the WRMD parameter bit [25] is asserted with bit [23] is set to 1. The WRMD parameter will specify which mode register and which chip selects (one or all) are to be programmed. The user is expected to define mode register write data accurately. If the mode register number specified in WRMD bits [7:0] is not a valid setting for the memory system being used, the memory controller may exhibit unpredictable behavior.

34.5.53 Control Register 52 (DDRMC_CR52)

Address: 400A_E000h base + D0h offset = 400A_E0D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								MR17_DA0								MR16_DA0								MR8_DA0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR52 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
23–16 MR17_DA0	Holds the memory mode register 17 data for chip select 0 written during memory initialization or when the WRMD bit [25] is asserted with bit [16] or bit [18] set to 1. The meaning of this parameter depends on the memory system being used: • For LPDDR2 memories: (MR17) for LPDDR2 S4 memories. If the lpddr2_s4 parameter is set to 1, the fields of MR17 will be programmed into the DRAM at initialization or when the WRMD is set to 1. • For DDR3 memories: This parameter has no meaning for this memory type.
15–8 MR16_DA0	Holds the memory mode register 16 data for chip select 0 written during memory initialization or when the WRMD bit [25] is asserted with bit [16] or bit [18] set to 1. The meaning of this parameter depends on the memory system being used: • For LPDDR2 memories: (MR16) The fields of MR16 will be programmed into the DRAM at initialization or when the WRMD bit is set to 1. • For DDR3 memories: This parameter has no meaning for this memory type.
7–0 MR8_DA0	Holds the memory mode register 8 (MR8) data for LPDDR2 memories for chip select 0 read during memory initialization. These parameters are read-only. The meaning of this parameter depends on the memory system being used: • For LPDDR2 memories: (MR8) The fields of MR8 will be read from the memories (LPDDR2-SDRAM) at initialization if the no_auto_mrr_init parameter is cleared to 0 or when the RMD bits are set to 1. • For DDR3 memories: This parameter has no meaning for this memory type.

34.5.54 Control Register 53 (DDRMC_CR53)

Address: 400A_E000h base + D4h offset = 400A_E0D4h

Bit	31–16	15–0
R	0	0
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR53 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 Reserved	This read-only field is reserved and always has the value 0.

34.5.55 Control Register 54 (DDRMC_CR54)

Address: 400A_E000h base + D8h offset = 400A_E0D8h

Bit	31–16	15–0
R	0	0
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR54 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 Reserved	This read-only field is reserved and always has the value 0.

34.5.56 Control Register 55 (DDRMC_CR55)

Address: 400A_E000h base + DCh offset = 400A_E0DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR55 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 Reserved	This read-only field is reserved and always has the value 0.

34.5.57 Control Register 56 (DDRMC_CR56)

Address: 400A_E000h base + E0h offset = 400A_E0E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR56 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–0 Reserved	This read-only field is reserved and always has the value 0.

34.5.58 Control Register 57 (DDRMC_CR57)

Address: 400A_E000h base + E4h offset = 400A_E0E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						CTRL_RAW		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR57 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 CTRL_RAW	Controls ECC error reporting (single bit and double bit errors) and correcting (single bit errors). **NOTE:** Before using the ECC mode the memory needs to be initialized such that all ECC bits are computed and stored in memory. 00 ECC not being used. 01 ECC reporting is on, but no attempts to correct. 10 No ECC RAM storage available. 11 ECC reporting and correcting on.
23–0 Reserved	This read-only field is reserved and always has the value 0.

34.5.59 Control Register 58 (DDRMC_CR58)

Address: 400A_E000h base + E8h offset = 400A_E0E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ECC_DIS_WCRER	0		XOR_CHK_BITS					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	XOR_CHK_BITS								0							
W																FWC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR58 field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value 0.
24 ECC_DIS_ WCRER	Disables automatic corruption of the ECC codes for an entire user word when the read portion of a read/modify/write operation has an un-correctable error. If this is not disabled, the corruption will occur even if the erroneous byte is overwritten with new data. 0 Allow the ECC codes for the entire user word to be corrupted. (Default) 1 Disable the corruption. The ECC codes written to memory will match the new write data written to memory.
23–22 Reserved	This read-only field is reserved and always has the value 0.
21–8 XOR_CHK_BITS	When the fwc parameter is set, the check bits generated by the next write operation will be XOR'ed with this parameter. The result will be written into memory as the new check value.

Table continues on the next page...

DDRMC_CR58 field descriptions (continued)

Field	Description
	• xor_chk_bits [13:7] maps to user word [63:32] • xor_chk_bits [6:0] maps to user word [31:0]
7–1 Reserved	This read-only field is reserved and always has the value 0.
0 FWC	Forces a write check. When this bit is set, the memory controller will XOR the xor_check_bits parameter with the generated checksum bits from the next word to be written to the memory. This parameter is write-only and will always read back as 0x0. Once the next write operation has been completed, the memory controller will automatically clear this bit to 0. 0 No action. 1 Force a write check.

34.5.60 Control Register 59 (DDRMC_CR59)

Address: 400A_E000h base + ECh offset = 400A_E0ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECC_U_ADR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR59 field descriptions

Field	Description
31–0 ECC_U_ADR	Holds the address of the read data that caused a double bit un-correctable ECC event. The memory controller will pad this parameter with zeros for any address bits not used by the memory controller. This parameter will only be used when an un-correctable error occurs and ECC error reporting is enabled in the ctrl_raw parameter. This parameter is read-only.

34.5.61 Control Register 60 (DDRMC_CR60)

Address: 400A_E000h base + F0h offset = 400A_E0F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									ECC_U_SYND						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR60 field descriptions

Field	Description
31–7 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR60 field descriptions (continued)

Field	Description
6–0 ECC_U_SYND	Stores the syndrome bits associated with a double bit un-correctable ECC error event. This parameter will only be used when an un-correctable error occurs and ECC error reporting is enabled in the ctrl_raw parameter. This memory controller can indicate that only 2 bits of the data and/or check code are erroneous but can not identify which bits. This parameter is read-only.

34.5.62 Control Register 61 (DDRMC_CR61)

Address: 400A_E000h base + F4h offset = 400A_E0F4h

Bit	31–0
R	ECC_U_DATA
W	
Reset	0000 0000 0000 0000 0000 0000 0000 0000

DDRMC_CR61 field descriptions

Field	Description
31–0 ECC_U_DATA	Contains the data associated with a double bit un-correctable ECC event. This parameter will only be used when an un-correctable error occurs and ECC error reporting is enabled in the ctrl_raw parameter.

34.5.63 Control Register 62 (DDRMC_CR62)

Address: 400A_E000h base + F8h offset = 400A_E0F8h

Bit	31–0
R	ECC_C_ADDR
W	
Reset	0000 0000 0000 0000 0000 0000 0000 0000

DDRMC_CR62 field descriptions

Field	Description
31–0 ECC_C_ADDR	Holds the address of the read data that caused a single bit correctable ECC event. The memory controller will pad this parameter with zeros for any address bits not used by the memory controller. This parameter will only be used when a correctable error occurs and ECC error reporting is enabled in the ctrl_raw parameter.

34.5.64 Control Register 63 (DDRMC_CR63)

Address: 400A_E000h base + FCh offset = 400A_E0FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										ECC_C_SYND					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR63 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 ECC_C_SYND	Stores the syndrome value associated with a single bit correctable ECC error event. This parameter will only be used when a correctable error occurs and ECC error reporting is enabled in the ctrl_raw parameter. This value will indicate which bit of the check code or data was erroneous.

34.5.65 Control Register 64 (DDRMC_CR64)

Address: 400A_E000h base + 100h offset = 400A_E100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECC_C_DATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR64 field descriptions

Field	Description
31–0 ECC_C_DATA	Contains the pre-corrected data associated with a single bit correctable ECC event. This parameter will only be used when a correctable error occurs and ECC error reporting is enabled in the ctrl_raw parameter.

34.5.66 Control Register 65 (DDRMC_CR65)

Address: 400A_E000h base + 104h offset = 400A_E104h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		ECC_C_ID														0			ECC_U_ID												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR65 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–16 ECC_C_ID	Contains the source ID associated with a single bit correctable ECC event. This parameter will only be used when a correctable error occurs and ECC error reporting is enabled in the ctrl_raw parameter.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–0 ECC_U_ID	Contains the source ID associated with a double bit un-correctable ECC event. This parameter will only be used when an un-correctable error occurs and ECC error reporting is enabled in the ctrl_raw parameter.

34.5.67 Control Register 66 (DDRMC_CR66)

Address: 400A_E000h base + 108h offset = 400A_E108h

Bit	31–28	27–16	15–12	11–0
R	0	ZQCL	0	ZQINIT
W				
Reset	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR66 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 ZQCL	Specifies the duration of wait time, in memory clocks, required for the memories to complete a long ZQ calibration command (ZQCL). This parameter is only applicable when the MC is programmed for the following memory systems: • LPDDR2 (dram_class = ’b0101) • DDR3 (dram_class = ’b0110)
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 ZQINIT	Specifies the duration of wait time, in memory clocks, required for the memories to complete a ZQ command during initialization. This parameter is only applicable when the MC is programmed for the following memory systems: • LPDDR2 (dram_class = ’b0101) • DDR3 (dram_class = ’b0110)

34.5.68 Control Register 67 (DDRMC_CR67)

Address: 400A_E000h base + 10Ch offset = 400A_E10Ch

Bit	31–28	27–16	15–12	11–0
R	0	0	0	ZQCS
W				
Reset	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR67 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 ZQCS	Specifies the duration of wait time, in memory clocks, required for the memories to complete a short ZQ calibration command (ZQCS). This parameter is only applicable when the MC is programmed for the following memory systems: • LPDDR2 (dram_class = ’b0101) • DDR3 (dram_class = ’b0110)

34.5.69 Control Register 68 (DDRMC_CR68)

Address: 400A_E000h base + 110h offset = 400A_E110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR68 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.70 Control Register 69 (DDRMC_CR69)

Address: 400A_E000h base + 114h offset = 400A_E114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				ZQ_ON_SREF_EX				0							
W																													ZQ_REQ			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR69 field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR69 field descriptions (continued)

Field	Description
11–8 ZQ_ON_SREF_EX	Issues a ZQ command when exiting selfrefresh mode. This is a one-hot parameter which determines the type of operation performed on self-refresh exit. **NOTE:** It is recommended to set this parameter to execute a ZQCL on selfrefresh exit. By setting this parameter this way, a ZQCL will be issued before any other commands are permitted to execute. If this parameter is programmed to any other setting, the system must manage self-refresh exit and ZQ calibration. If using the ZQCS option, it is advised that the rotate function should be disabled (the parameter zqcs_rotate should be cleared to 0) to calibrate all chip selects. • Bit [3] — Triggers a ZQ reset to all chip selects in the system on self-refresh exit. (This bit is only applicable for LPDDR2 memory systems) • Bit [2] — Triggers a ZQ initialization to all chip selects in the system on selfrefresh exit. (This bit is only applicable for LPDDR2 memory systems) • Bit [1] — Triggers a long ZQ calibration to all chip selects in the system on selfrefresh exit. • Bit [0] — Triggers a short ZQ calibration. If the zqcs_rotate parameter is set to 1, only one chip select will be calibrated (in round-robin fashion). If the zqcs_rotate parameter is cleared to 0, all chip selects defined in the parameter cs_map will be calibrated. This parameter is only applicable when the MC is programmed for the following memory systems: • LPDDR2 (dram_class = ’b0101) • DDR3 (dram_class = ’b0110)
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 ZQ_REQ	Triggers a user-requested ZQ operation. This parameter is write-only and will always read back as 0x0. • Bit [3] — Triggers a ZQ reset to all chip selects in the system. (This bit is only applicable for LPDDR2 memory systems) • Bit [2] — Triggers a ZQ initialization to all chip selects in the system. (This bit is only applicable for LPDDR2 memory systems) • Bit [1] — Triggers a long ZQ calibration to all chip selects in the system. • Bit [0] — Triggers a short ZQ calibration. This parameter is only applicable when the MC is programmed for the following memory systems: • LPDDR2 (dram_class = ’b0101) • DDR3 (dram_class = ’b0110)

34.5.71 Control Register 70 (DDRMC_CR70)

Address: 400A_E000h base + 118h offset = 400A_E118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	REF_PER_ZQ																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR70 field descriptions

Field	Description
31–0 REF_PER_ZQ	Sets the maximum number of refreshes allowed between automatic ZQCS commands. Setting this parameter to 0x0 will disable automatic ZQCS commands. When using the rotate option (the parameter zqcs_rotate is set to 1), this parameter should be programmed to the total number of controller clocks desired between ZQCS commands divided by the number of chip selects in the system.

DDRMC_CR70 field descriptions (continued)

Field	Description
	This parameter is only applicable when the MC is programmed for the following memory systems: • LPDDR2 (dram_class = ’b0101) • DDR3 (dram_class = ’b0110)

34.5.72 Control Register 71 (DDRMC_CR71)

Address: 400A_E000h base + 11Ch offset = 400A_E11Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												ZQRESET			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ZQRESET								0							ZQ_IN_PROG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR71 field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–8 ZQRESET	Specifies the duration of wait time, in memory clocks, required for the memories to complete a ZQRESET command.

Table continues on the next page...

DDRMC_CR71 field descriptions (continued)

Field	Description
	This parameter is only applicable when the MC is programmed for the following memory systems: LPDDR2 (dram_class = ’b0101)
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 ZQ_IN_PROG	Indicates that a ZQ command is currently in progress. If a ZQ command is requested while this parameter is set to 1, the new ZQ request will be ignored. This parameter is read-only. • 0 — Not in progress • 1 — In progress

34.5.73 Control Register 72 (DDRMC_CR72)

Address: 400A_E000h base + 120h offset = 400A_E120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ZQCS_ROTATE	0							NO_ZQ_INIT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR72 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 ZQCS_ROTATE	Defines the behavior of short ZQ calibrations. If this parameter is set to 1, a ZQCS request will only be issued to one chip select on each request, regardless of the source of the request. The MC will maintain a counter such that the chip selects are calibrated in a fixed rotating sequence. 0 = Calibrate all chip selects on each ZQCS request. 1 = Calibrate only one chip select on the next ZQCS request. This parameter is only applicable when the MC is programmed for the following memory systems: • LPDDR2 (dram_class = ’b0101) • DDR3 (dram_class = ’b0110)
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR72 field descriptions (continued)

Field	Description
16 NO_ZQ_INIT	Disables ZQ operations during initialization. 0 = ZQ operations allowed during initialization. 1 = ZQ operations disabled during initialization. This parameter is only applicable when the MC is programmed for the following memory systems: • LPDDR2 (dram_class = 'b0101)
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.74 Control Register 73 (DDRMC_CR73)

Address: 400A_E000h base + 124h offset = 400A_E124h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				APREBIT				0					COL_DIFF		
W																
Reset	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						ROW_DIFF		0						BANK_DIFF	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR73 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 APREBIT	Defines the location of the auto precharge bit in the DRAM address in decimal encoding. **NOTE:** This parameter will be cleared to 0x0 on reset, but will change to the default value on the first controller clock edge after reset.
23–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 COL_DIFF	Shows the difference between the maximum column width available (11) and the actual number of column pins being used. The user address is automatically shifted so that the user address space is mapped contiguously into the memory map based on the value of this parameter.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 ROW_DIFF	Defines the difference between the maximum number of address pins configured (16) and the actual number of pins being used. **NOTE:** For DDR3 memory systems (dram_class = 'b0110), this parameter should reflect the difference in the number of address pins configured and the number of row pins being used. **NOTE:** For LPDDR2 memory systems (dram_class = 'b0101), this parameter holds the difference between the maximum number of row pins configured and the actual number of row pins being used for chip select X.

Table continues on the next page...

DDRMC_CR73 field descriptions (continued)

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 BANK_DIFF	Defines the difference between the maximum number of banks configured (3) and the actual number of banks being used. 00 All bank bits are being used (8 banks) 01 Only 2 bank bits are being used (4 banks) 10 Only 1 bank bit is being used (2 banks) 11 Reserved

34.5.75 Control Register 74 (DDRMC_CR74)

Address: 400A_E000h base + 128h offset = 400A_E128h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							BANKSPLT_EN	0							ADDR_CMP_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	CMD_AGE_CNT								AGE_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR74 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 BANKSPLT_EN	Enables bank splitting as a condition when using the placement logic to fill the command queue. 0 Disable 1 Enable
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 ADDR_CMP_EN	Enables address collision/data coherency detection as a condition when using the placement logic to fill the command queue. 0 Diasble 1 Enable

Table continues on the next page...

DDRMC_CR74 field descriptions (continued)

Field	Description
15–8 CMD_AGE_CNT	Holds the initial value of the command aging counters associated with each command in the command queue. When using the placement logic to fill the command queue, the command aging counters decrement one each time the master aging-rate counter counts down the number of cycles in the age_count parameter. This parameter is only applicable when priority is enabled as a placement factor (the parameter is set to 1).
7–0 AGE_CNT	Holds the initial value of the master aging-rate counter. This parameter is only applicable when priority is enabled as a placement factor

34.5.76 Control Register 75 (DDRMC_CR75)

Address: 400A_E000h base + 12Ch offset = 400A_E12Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							RW_PG_EN	0							RW_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							PRI_EN	0							PLEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR75 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 RW_PG_EN	Enables page grouping within a read/ write group as a condition when using the placement logic to fill the command queue. 0 Disable 1 Enable
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 RW_EN	Enables read/write grouping as a condition when using the placement logic to fill the command queue. 0 Disable 1 Enable
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR75 field descriptions (continued)

Field	Description
8 PRI_EN	Enables priority as a condition when using the placement logic to fill the command queue. 0 Disable 1 Enable
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 PLEN	Enables using the placement logic to fill the command queue. 0 Disable placement logic. The command queue is a straight FIFO. 1 Enable placement logic. The command queue will be filled according to the placement logic factors.

34.5.77 Control Register 76 (DDRMC_CR76)

Address: 400A_E000h base + 130h offset = 400A_E130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					NQENT_ACTDIS			0						D_RW_G_BKCN	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							W2R_SPLT_EN	0							CS_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR76 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 NQENT_ACTDIS	Specifies the number of entries of the command queue in which ACT requests are not allowed. For this 8-deep command queue, the entries are numbered 0-7, where entry 0 is the command next to execute. • 0x0 = ACT allowed from entries 0-7 • 0x1 = ACT allowed from entries 0-6 • 0x2 = ACT allowed from entries 0-5 • 0x3 = ACT allowed from entries 0-4 • 0x4 = ACT allowed from entries 0-3 • 0x5 = ACT allowed from entries 0-2

Table continues on the next page...

DDRMC_CR76 field descriptions (continued)

Field	Description
	• 0x6 = ACT allowed from entries 0-1 • 0x7 = ACT allowed from entry 0
23–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 D_RW_G_BKCN	Enables read/write grouping as a condition when using the placement logic to fill the command queue. • Bit [1] = Prohibits placement into a command queue entry 2 entries away (before or after) from a command with a bank conflict. • 0 — Allowed • 1 — Prohibited • Bit [0] = Prohibits placement into a command queue entry immediately before or immediately after a command with a bank conflict. • 0 — Allowed • 1 — Prohibited **NOTE:** It is not meaningful to set bit [1] of this parameter without setting bit [0].
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 W2R_SPLT_EN	Enables write-to-read turnaround time optimization for commands to the same chip select as a condition when using the placement logic to fill the command queue. When this parameter is set to 1, the placement logic will attempt to place commands between two entries that are accessing the same chip select, but are different types, and where the write is going to execute before the read. 0 Disable 1 Enable
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 CS_EN	Enables rank grouping within a R/W group as a condition when using the placement logic to fill the command queue. If the rw_same_en parameter is set to 1, the placement logic will try to group commands of the same type and same rank when possible. If there are no rank grouping opportunities, the logic will still attempt to group commands of the same type. If the rw_same_en parameter is cleared to 0, this parameter has no effect. 0 Disable 1 Enable

34.5.78 Control Register 77 (DDRMC_CR77)

Address: 400A_E000h base + 134h offset = 400A_E134h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							CS_MAP	0						IN_DRAM_CMD	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							DI_RD_INTLEAVE	0							SWAP_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR77 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 CS_MAP	Set this bit to enable the CS
23–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 IN_DRAM_CMD	Inhibits certain types of commands from being executed from the command queue. Even when this parameter is programmed to a non-zero value, commands may still be accepted into the memory controller and the core command queue, but they will not be executed. • 00 = Enable any command in the command to execute. • 01 = Inhibit read/write traffic and associated bank commands in the command queue from being executed. • 10 = Inhibit MRR and peripheral MRR commands from being executed. • 11 = Inhibit MRR commands and read/write commands in the command queue from being executed. The MRR settings are only applicable when the MC is programmed for the following memory systems: LPDDR2 (dram_class = ’b0101)
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 DI_RD_ INTLEAVE	Disables read data interleaving for commands from the same port, regardless of the requestor ID. Read data may still be returned out of order regardless of the value of this parameter. 0 Allow read data interleaving 1 Disable read data interleaving

Table continues on the next page...

DDRMC_CR77 field descriptions (continued)

Field	Description
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 SWAP_EN	Enables swapping of the active command for a new higher-priority command when using the placement logic. 0 Disable 1 Enable

34.5.79 Control Register 78 (DDRMC_CR78)

Address: 400A_E000h base + 138h offset = 400A_E138h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					Q_FULLNESS			0							LPDDR2_S4
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							REDUC	0				BUR_ON_FLY_BIT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR78 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 Q_FULLNESS	Defines quantity of data that will be considered full for the command queue.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 LPDDR2_S4	Indicates the type of LPDDR2 memory(s) being used. The user must manually set this bit for LPDDR2-S4 memory(s). • 0 = LPDDR2-S2 memory in the system • 1 = LPDDR2-S4 memory in the system This parameter is only applicable when the MC is programmed for the following memory systems: LPDDR2 (dram_class = 'b0101)
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR78 field descriptions (continued)

Field	Description
8 REDUC	Allows the same memory controller to be used with memories with a smaller datapath. When enabled, certain bits of the PHY data bus are unused since the memory data bus is half the width of the initial configured size. **NOTE:** The entire user datapath is used regardless of this setting. **NOTE:** This parameter should not be changed after the start parameter has been set to 1. • 0 = Standard operation using full memory bus. • 1 = Memory datapath width is half of the maximum size.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 BUR_ON_FLY_BIT	Defines the bit of the DRAM address that defines Burst-On-Fly behavior. The memory controller does not support burst-on-fly behavior at this time. However, this parameter must be programmed to ’b1100 for accurate system operation as defined by the JEDEC specification. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)

34.5.80 Control Register 79 (DDRMC_CR79)

Address: 400A_E000h base + 13Ch offset = 400A_E13Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							CTLUPD_AREF	0							
W																CTLUPD_REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CTLBUSY	0							INODR_ACT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR79 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 CTLUPD_AREF	Enables an automatic controller-initiated update after every refresh. 0 Disable 1 Enable
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR79 field descriptions (continued)

Field	Description
16 CTLUPD_REQ	Triggers a controller-initiated update request on the PHY interface. This parameter is write-only and will always read back as 0x0. **NOTE:** For proper operation, the user must issue at least one read or write command to memory after reset prior to setting this parameter. 0 No action 1 Trigger a controller-initiate update request
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 CTLBUSY	Indicates that the memory controller is actively processing a command from any port. This information is also passed to the ASIC through the controller_busy signal. This parameter is read-only. 0 MC is not busy 1 MC is busy
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 INODR_ACT	Forces the controller to accept commands in the order in which they are placed in the command queue.

34.5.81 Control Register 80 (DDRMC_CR80)

Address: 400A_E000h base + 140h offset = 400A_E140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			INT_STAT																												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR80 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–0 INT_STAT	Shows the status of all possible interrupts generated by the controller. **NOTE:** If possible, the location of the int_stat parameter has been preserved from the previous delivery. However, the individual bits, their meaning and the size of the parameter may have changed. The int_stat bits correspond to these interrupts: • Bit [28] = Logical OR of all lower bits. • Bit [27] = The frequency change hardware has completed all operations. The system will hold off normal operations if the FRQ_CHG_HOEN parameter is set to 1. • Bit [26] = The PHY tINIT_COMPLETE value has timed out. This value is specified in the PHY_INI_COM parameter. • Bit [25] = The user-initiated DLL resync has completed. • Bit [24] = A state change has been detected on the dfi_init_complete signal after initialization. • Bit [23] = The assertion of the IN_DRAM_CMD parameter has successfully inhibited the command queue and/or MRR traffic.

Table continues on the next page...

DDRMC_CR80 field descriptions (continued)

Field	Description
	• Bit [22] = The register interface-initiated mode register write has completed and another mode register write may be issued. • Bit [21] = A temperature alert condition (low or high temp) has been detected. • Bit [20] = The last automatic MRR of MR4 indicated a change in the device temperature or refresh rate (TUF bit set). • Bit [19] = The requested mode register read has completed. The chip and data can be read in the PERI_MRR_DA parameter. • Bit [18] = The leveling operation has completed. • Bit [17] = A leveling operation has been requested. • Bit [16] = A DFI update error has occurred. Error information can be found in the UP_ERR_STAT parameter. • Bit [15] = A write leveling error has occurred. Error information can be found in the WRLVL_ERR_STAT parameter. • Bit [14] = A read leveling gate training error has occurred. Error information can be found in the RDLVL_ERROR_STATUS parameter. • Bit [13] = A read leveling error has occurred. Error information can be found in the RDLV_ERR_STAT parameter. • Bit [12] = ODT has been enabled while the MC is programmed for CAS latency 3. This is an unsupported combination. • Bit [11] = The user has programmed an invalid setting associated with user words per burst. Examples: Setting param_reduc when burst length = 2. A 1:2 MC:PHY clock ratio with burst length = 2. • Bit [10] = A wrap cycle crossing a DRAM page has been detected. This is unsupported and may result in memory data corruption. • Bit [9] = The low power operation has been completed. • Bit [8] = The MC initialization has been completed. • Bit [7] = An error occurred on the port command channel. • Bit [6] = Multiple uncorrectable ECC events have been detected. • Bit [5] = An uncorrectable ECC event has been detected. • Bit [4] = Multiple correctable ECC events have been detected. • Bit [3] = A correctable ECC event has been detected. • Bit [2] = Multiple accesses outside the defined PHYSICAL memory space have occurred. • Bit [1] = A memory access outside the defined PHYSICAL memory space has occurred. • Bit [0] = The memory reset is valid on the DFI bus.

34.5.82 Control Register 81 (DDRMC_CR81)

Address: 400A_E000h base + 144h offset = 400A_E144h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W					INT_ACK																											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR81 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR81 field descriptions (continued)

Field	Description
27–0 INT_ACK	Clear mask of the INT_STAT parameter. Controls the clearing of the int_stat parameter. This parameter is write-only and will always read back as 0x0. 0 No effect 1 Clears the associated bit in the int_stat parameter to 0.

34.5.83 Control Register 82 (DDRMC_CR82)

Address: 400A_E000h base + 148h offset = 400A_E148h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			INT_MASK																												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR82 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

28–0
INT_MASK
	Mask for controller_int signals from the INT_STAT parameter.
	
	Active-high mask bits that control the value of the memory controller_int signal on the ASIC interface.
	• Bit [28] = Masks all interrupt reporting.
	  • 0 = Mask interrupts individually based on the settings of bits [27:0] of this parameter.
	  • 1 = Mask all interrupts. The settings in bits [27:0] of this parameter are not relevant.
	• Bits [27:0] = Individual mask bits for each interrupt in the int_stat parameter. For each bit:
	  • 0 = Do not mask interrupt
	  • 1 = Mask interrupt. This will prevent a set interrupt from causing the controller_int signal to be asserted.

34.5.84 Control Register 83 (DDRMC_CR83)

Address: 400A_E000h base + 14Ch offset = 400A_E14Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OORAD																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR83 field descriptions

Field	Description
31–0 OORAD	Address of command that caused an out-of-range interrupt. Holds the address of the command that caused either of the out-of-range interrupts (bits 1 or 2) in the int_stat parameter to be set to 1.

34.5.85 Control Register 84 (DDRMC_CR84)

Address: 400A_E000h base + 150h offset = 400A_E150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		OORID													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		OORTYP						0	OORLEN						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR84 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–16 OORID	Source ID of command that caused an out-of-range interrupt Holds the source ID of the command that caused either of the out-of-range interrupts (bits 1 or 2) in the int_stat parameter to be set to 1.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–8 OORTYP	Type of command that caused an out-of-range interrupt. Holds the type of command that caused either of the out-of-range interrupts (bits 1 or 2) in the int_stat parameter to be set to 1.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 OORLEN	Length of command that caused an out-of-range interrupt. Holds the length of the command that caused either of the out-of-range interrupts (bits 1 or 2) in the int_stat parameter to be set to 1.

34.5.86 Control Register 85 (DDRMC_CR85)

Address: 400A_E000h base + 154h offset = 400A_E154h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	P_CMDERRADD																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR85 field descriptions

Field	Description
31–0 P_CMDERRADD	Address of command that caused the PORT command error Holds the address of the command that caused a port command error condition

34.5.87 Control Register 86 (DDRMC_CR86)

Address: 400A_E000h base + 158h offset = 400A_E158h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														P_CMDERR_TYP	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		P_CMDERRID													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR86 field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 P_CMDERR_TYP	Type of error and access type that caused the PORT command error Defines the type of error and the access type that caused the port command error condition. If multiple bits are set to 1, then multiple errors were found.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–0 P_CMDERRID	Source ID of command that caused the PORT command error

34.5.88 Control Register 87 (DDRMC_CR87)

Address: 400A_E000h base + 15Ch offset = 400A_E15Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ODT_WR_ MAPCS0	0							ODT_RD_ MAPCS0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR87 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 ODT_WR_ MAPCS0	Determines which chip(s) will have termination when a write occurs on chip select 0. Sets up which (if any) chip(s) will have their ODT termination active while a write occurs on chip select 0. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110). 0 CS0 will have active ODT termination when chip select 0 is performing a write.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 ODT_RD_ MAPCS0	Determines which chip(s) will have termination when a read occurs on chip select 0. Sets up which (if any) chip(s) will have their ODT termination active while a read occurs on chip select 0. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110) 0 CS0 will have active ODT termination when chip select 0 is performing a read
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.89 Control Register 88 (DDRMC_CR88)

Address: 400A_E000h base + 160h offset = 400A_E160h

Bit	31–21	20–16	15–0
R	0	TODTL_CMD	0
W			
Reset	0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR88 field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 TODTL_CMD	DRAM delay requirement from ODT de-assertion to next nonwrite, non-read command. Defines the DRAM timing between an ODT to the next command, in memory clocks • For LPDDR2 memories: This parameter has no meaning for this memory type • For DDR3 memories: Program to the (ODTL_off + 1) value from the memory specification.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.90 Control Register 89 (DDRMC_CR89)

Address: 400A_E000h base + 164h offset = 400A_E164h

Bit	31–12	11–8	7–4	3–0
R	0	AODT_WRSMCS	0	AODT_RWSMCS
W				
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

DDRMC_CR89 field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 AODT_WRSMCS	Additional delay to insert between write and read transaction types to the same chip select to meet ODT timing requirements. Any value including 0x0 supported. Defines the number of additional controller clocks of delay to insert after a write command before a read command to the same chip select to meet ODT timing requirements.This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110)
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 AODT_RWSMCS	Additional delay to insert between read and write transaction types to the same chip select to meet ODT timing requirements. Defines the number of additional controller clocks of delay to insert after a read command before a write command to the same chip select to meet ODT timing requirements.This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110).

34.5.91 Control Register 90 (DDRMC_CR90)

Address: 400A_E000h base + 168h offset = 400A_E168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR90 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.92 Control Register 91 (DDRMC_CR91)

Address: 400A_E000h base + 16Ch offset = 400A_E16Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					W2R_ SMCSDL			0					R2W_ SMCSDL			0					R2R_ SMCSDL			0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR91 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–24 W2R_SMCSDL	Additional delay to insert between writes and reads to the same chip select. Defines the number of additional controller clocks of delay to insert from a write command to a read command to the same chip select. This parameter may be programmed to any value including 0x0.
23–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 R2W_SMCSDL	Additional delay to insert between reads and writes to the same chip select. Program to a non-zero value. Defines the number of additional controller clocks of delay to insert from a read command to a write command to the same chip select. This parameter should be programmed to a non-zero value. The exact value is system-dependent.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 R2R_SMCSDL	Additional delay to insert between two reads to the same chip select. Any value including 0x0 supported. Defines the number of additional controller clocks of delay to insert between two read commands to the same chip select. This parameter may be programmed to any value including 0x0.

Table continues on the next page...

DDRMC_CR91 field descriptions (continued)

Field	Description
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.93 Control Register 92 (DDRMC_CR92)

Address: 400A_E000h base + 170h offset = 400A_E170h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														TDQSCK_MIN	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						TDQSCK_MAX		0					W2W_SMCSDL		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR92 field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 TDQSCK_MIN	Additional delay needed for tDQSCK. Defines the minimum value of the timing term tDQSCK, in memory clocks. This holds the delay between the memory clock (read latency cycles after the read command) and the first read DQS rising edge. This parameter is only applicable when the MC is programmed for the following memory systems: LPDDR2 (dram_class = 0101).
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 TDQSCK_MAX	Additional delay needed for tDQSCK. Defines the maximum value of the timing term tDQSCK, in memory clocks. This holds the delay between the memory clock (read latency cycles after the read command) and the first read DQS rising edge. This parameter is only applicable when the MC is programmed for the following memory systems:LPDDR2 (dram_class = 0101).
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 W2W_SMCSDL	Additional delay to insert between two writes to the same chip select. Any value including 0x0 supported. Defines the number of additional controller clocks of delay to insert between two write commands to the same chip select. This parameter may be programmed to any value including 0x0.

34.5.94 Control Register 93 (DDRMC_CR93)

Address: 400A_E000h base + 174h offset = 400A_E174h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							
W								SWLVL_START								SWLVL_LOAD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						SW_LVL_ MODE		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR93 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 SWLVL_START	User request to initiate software leveling of type in the SW_LVL_MODE bits. Set to 1 to trigger. Initiates the software leveling operation defined in the SW_LVL_MODE bits. This parameter is write-only and will always read back as 0x0. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 'b0110) 0 No Action 1 Initiate software leveling operation
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 SWLVL_LOAD	User request to load delays and execute software leveling. Set to to trigger.

Table continues on the next page...

DDRMC_CR93 field descriptions (continued)

Field	Description
	Triggers the delays to be loaded into the PHY delay lines and initiate a read burst or write strobe for software leveling. This parameter is write-only and will always read back as 0x0. This parameter is only applicable when the MC is programmed for the following memory systems:DDR3 (dram_class = 0110) 0 No action 1 Load delays and start software leveling
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 SW_LVL_MODE	Defines the leveling operation for software leveling. Set to 0 for none, set to 1 for write leveling, set to 2 for read leveling, or set to 3 for gate training. Defines the type of leveling operation performed during the next software leveling operation. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110) 00 No Leveling 01 Write Leveling 10 Read Leveling 11 Gate Training
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.95 Control Register 94 (DDRMC_CR94)

Address: 400A_E000h base + 178h offset = 400A_E178h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SWLVL_RESP_0								0					LVL_STATUS		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							SWLVL_OP_DONE	0							
W																SWLVL_EXIT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR94 field descriptions

Field	Description
31–24 SWLVL_RESP_0	Leveling response for data slice 0. Response for the software leveling process for data slice 0. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice.This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110).

Table continues on the next page...

DDRMC_CR94 field descriptions (continued)

Field	Description
23–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 LVL_STATUS	Status of write level, read level, and gate training requests. This is used with the LVL_REQ interrupt. Bit (0) correlates to write leveling request, bit (1) correlates to read leveling request, and bit (2) correlates to gate training request. Value of 1indicates request received. Indicates the type of leveling request that resulted in the leveling request interrupt (bit 16) in the int_stat parameter being set to 1. This interrupt will be triggered if any of the PHY request signals are asserted. **NOTE:** For software leveling, the user may ignore this parameter and the lvl_status parameter and trigger a software leveling request through the swlvl_start parameter directly. • Bit [2] = Gate training request status • Bit [1] = Read leveling request status • Bit [0] = Write leveling request status For all bits : • 0 = No request received • Request received through the PHY interface, the internal refresh counters or the register interface. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110)
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 SWLVL_OP_ DONE	Signals that software leveling is currently in progress. Value of 1 indicates operation complete. Reports on status of the software leveling operation. This parameter will be driven low during initiation, load or exit operations. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110) 0 Operation in process 1 Operation completed
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 SWLVL_EXIT	User request to exit software leveling. Set to 1 to exit. User request to exit software leveling. This parameter is write-only and will always read back as 0x0. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110) 0 No action 1 Exit software leveling

34.5.96 Control Register 95 (DDRMC_CR95)

Address: 400A_E000h base + 17Ch offset = 400A_E17Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							WRLVL_CS	0							
W																WRLVL_REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SWLVL_RESP_2								SWLVL_RESP_1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR95 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 WRLVL_CS	Specifies the target chip select for the write leveling operation. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110). 0 Targets CS0
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 WRLVL_REQ	User request to initiate write leveling. Set to 1 to trigger. User request to initiate write leveling. This parameter is write-only and will always read back as 0x0. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110). 0 No action 1 Trigger write leveling

Table continues on the next page...

DDRMC_CR95 field descriptions (continued)

Field	Description
15–8 SWLVL_RESP_2	Leveling response for data slice 2. Response for the software leveling process for data slice 2. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110)
7–0 SWLVL_RESP_1	Leveling response for data slice 1. Response for the software leveling process for data slice1 . Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110)

34.5.97 Control Register 96 (DDRMC_CR96)

Address: 400A_E000h base + 180h offset = 400A_E180h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															WRLVL_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		WLMRD						0		WLDQSEN					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR96 field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 WRLVL_EN	Enable the MC write leveling module. Set to 1 to enable. Enables the hardware write leveling feature in the memory controller.This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110). 0 Disable. The memory controller will never drive the PHY WRITE LEVEL ENABLE signal. The user may program the write leveling delay manually in the wrlvl_delay_X parameters. 1 Enable. Write leveling will automatically be run at initialization, may be run on demand by setting the PHY WRITE LEVEL ENABLE parameter to 1 to assert the signal, or may be run periodically based on the programming of the wrlvl_refresh_interval parameter.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR96 field descriptions (continued)

Field	Description
13–8 WLMRD	Delay from issuing MRS to first write leveling strobe. Defines the delay from when an MRS command is issued to when the first write leveling strobe is issued, in controller clocks. This is a write leveling-specific timing specification.This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110).
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 WLDQSEN	Delay from issuing MRS to first DQS strobe for write leveling. Minimum number of controller clocks of delay after memory enters write leveling mode (MRS command issued) before first DQS strobe can be sent. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110).

34.5.98 Control Register 97 (DDRMC_CR97)

Address: 400A_E000h base + 184h offset = 400A_E184h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							WRLVL_REG_EN	WRLVL_ERR_STAT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WRLVL_REFRESH_INTERVAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR97 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 WRLVL_REG_EN	Enable the PHY WRITE LEVEL DELAY X signals to be programmed when hardware and software leveling are disabled. Set to 1 to enable. Enables direct control of the PHY WRITE LEVEL DELAY X signals through the WRLVL_DL_X parameters. This is used when hardware and software leveling are both disabled.This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110) 0 Disable 1 Enable
23–16 WRLVL_ERR_STAT	Holds the status from the most recent write leveling operation. This parameter is read-only. • Bit [7] = Indicates that the leveling operation exceeded the maximum allowed enable time.This bit will only report errors if the PHY_WRLVL_RESP parameter is programmed to a non-zero value. • Bit [6] = Indicates that the leveling operation exceeded the maximum allowable response time (PHY WRITE LEVEL MAX). This bit will only report errors if the PHY_WRLVL_MAX parameter is programmed to a non-zero value. • Bits [5:0] = Reserved This parameter is used by the write leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110)
15–0 WRLVL_REFRESH_INTERVAL	Number of refreshes counted between automatic write leveling commands Sets the maximum number of refreshes allowed between automatic write leveling operations. Clearing this parameter to 0x0 will disable automatic write leveling. If this parameter is programmed to a nonzero value, refresh sequences will be counted and the leveling request interrupt (bit 16) in the int_stat parameter will be set to 1 when the counter expires. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 0110)

34.5.99 Control Register 98 (DDRMC_CR98)

Address: 400A_E000h base + 188h offset = 400A_E188h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																WRLVL_DL_0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR98 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 WRLVL_DL_0	Number of delay elements for write data slice 0. This is readonly if the WRLVL_EN parameter is 1. If both the WRLVL_EN and the WRLVL_REG_EN parameters are 0, this can only be programmed during initialization or through software leveling. If the WRLVL_EN parameter is 0 and the WRLVL_REG_EN parameter is 1, this can be programmed at any time.

34.5.100 Control Register 99 (DDRMC_CR99)

Address: 400A_E000h base + 18Ch offset = 400A_E18Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																WRLVL_DL_1															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR99 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 WRLVL_DL_1	Number of delay elements for write data slice 1. This is readonly if the WRLVL_EN parameter is 1. If both the WRLVL_EN and the WRLVL_REG_EN parameters are 0, this can only be programmed during initialization or through software leveling. If the WRLVL_EN parameter is 0 and the WRLVL_REG_EN parameter is 1, this can be programmed at any time.

34.5.101 Control Register 100 (DDRMC_CR100)

Address: 400A_E000h base + 190h offset = 400A_E190h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																WRLVL_DL_2															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR100 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 WRLVL_DL_2	Number of delay elements for write data slice 2. This is readonly if the WRLVL_EN parameter is 1. If both the WRLVL_EN and the WRLVL_REG_EN parameters are 0, this can only be programmed during initialization or through software leveling. If the WRLVL_EN parameter is 0 and the WRLVL_REG_EN parameter is 1, this can be programmed at any time.

34.5.102 Control Register 101 (DDRMC_CR101)

Address: 400A_E000h base + 194h offset = 400A_E194h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							RDLVL_EDGE	0							RDLVL_CS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0							
W								RDLVL_GT_REQ								RDLVL_REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR101 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 RDLVL_EDGE	Specifies the read DQS edge to be used for the read leveling operation. • 0 = Positive edge • 1 = Negative edge This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class*) = 0110
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR101 field descriptions (continued)

Field	Description
16 RDLVL_CS	Specifies the target chip select for read leveling. This information will be conveyed to the PHY. • 0 = targets CS0 This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class*) = 0110
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 RDLVL_GT_REQ	User request to initiate gate training. • 0 = No action • 1 = Trigger gate training This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 RDLVL_REQ	User request to initiate read leveling. Set to 1 to trigger.

34.5.103 Control Register 102 (DDRMC_CR102)

Address: 400A_E000h base + 198h offset = 400A_E198h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															RDLVL_GT_REGEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							RDLVL_REG_EN	0							RDLVL_BGN_DLEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR102 field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR102 field descriptions (continued)

Field	Description
16 RDLVL_GT_ REGEN	Enable the gate delay to be programmed when hardware and software leveling are disabled. Set to 1 to enable.
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 RDLVL_REG_EN	Enable the delay to be programmed when hardware and software leveling are disabled. Set to 1 to enable.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 RDLVL_BGN_ DLEN	Enables the hardware read leveling logic to begin finding the DQ data eye. If the process is not able to find the beginning of the data window, the delay will be cleared to 0 and the status bit will indicate the inability to locate the start. • 0 = Disable • 1 = Enable This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)

34.5.104 Control Register 103 (DDRMC_CR103)

Address: 400A_E000h base + 19Ch offset = 400A_E19Ch

Bit	31–16	15–0
R	RDLVL_END_DL0	RDLVL_BGN_DL0
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR103 field descriptions

Field	Description
31–16 RDLVL_END_ DL0	Indicates the value at which the first DQ transition (0 to 1) was found for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. These parameters are read-only. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)
15–0 RDLVL_BGN_ DL0	Indicates the value at which the first DQ transition (1 to 0) was found for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. These parameters are read-only. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)

34.5.105 Control Register 104 (DDRMC_CR104)

Address: 400A_E000h base + 1A0h offset = 400A_E1A0h

Bit	31–16	15–0
R	RDLVL_OFF_DL0	RDLVL_MP_DL0
W	RDLVL_OFF_DL0	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR104 field descriptions

Field	Description
31–16 RDLVL_OFF_ DL0	Offset for the read level midpoint delay for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)
15–0 RDLVL_MP_DL0	Indicates the calculated midpoint value of the DQ delay for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)

34.5.106 Control Register 105 (DDRMC_CR105)

Address: 400A_E000h base + 1A4h offset = 400A_E1A4h

Bit	31–16
R	0
W	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	15–8	7–1	0
R	RDLVL_DL_0	0	RDLVL_OFF_DIR_0
W	RDLVL_DL_0		RDLVL_OFF_DIR_0
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0

DDRMC_CR105 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR105 field descriptions (continued)

Field	Description
15–8 RDLVL_DL_0	Number of delay elements where read DQS is placed within the DQ data eye for data slice 0. This is read-only if the RDLVL_EN parameter is 1. If both the RDLVL_EN and the RDLVL_REG_EN parameters are 0, this can only be programmed during initialization or through software leveling. If the RDLVL_EN parameter is 0 and the RDLVL_REG_EN parameter is 1, this can be programmed at any time. **NOTE:** Value programmed in this field delays the strobe by (period of strobe/128) x the programmed value. For e.g, if the value programmed is 0x20h, the the strobe is delayed by period of strobe/4.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 RDLVL_OFF_ DIR_0	Specifies the direction of the offset for the read level midpoint delay for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. • 0 = Subtract delay • 1 = Add delay This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)

34.5.107 Control Register 106 (DDRMC_CR106)

Address: 400A_E000h base + 1A8h offset = 400A_E1A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								RDLVL_GTDL_0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR106 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 RDLVL_GTDL_0	Number of delay elements where gate is aligned to rising edge of DQS for data slice 0. This is readonly if the RDLVL_GATE_EN parameter is 1. If both the RDLVL_GATE_EN and the RDLVL_GATE_REG_EN parameters are 0, this can only be programmed during initialization or through software leveling. If the RDLVL_GATE_EN parameter is 0 and the RDLVL_GATE_REG_EN parameter is 1, this can be programmed at any time. **NOTE:** Value programmed in this field delays the strobe by (period of strobe/128)x the programmed value. For e.g, if the value programmed is 0x20h, the the strobe is delayed by period of strobe/4.

34.5.108 Control Register 107 (DDRMC_CR107)

Address: 400A_E000h base + 1ACh offset = 400A_E1ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RDLVL_BGN_DL1								0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR107 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 RDLVL_BGN_ DL1	Indicates the value at which the first DQ transition (1 to 0) was found for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. These parameters are read-only. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110) **NOTE:** Value programmed in this field delays the strobe by (period of strobe/128)x the programmed value. For e.g, if the value programmed is 0x20h, the the strobe is delayed by period of strobe/4.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.109 Control Register 108 (DDRMC_CR108)

Address: 400A_E000h base + 1B0h offset = 400A_E1B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDLVL_MP_DL1																RDLVL_END_DL1															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR108 field descriptions

Field	Description
31–16 RDLVL_MP_DL1	Indicates the calculated midpoint value of the DQ delay for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)
15–0 RDLVL_END_ DL1	Indicates the value at which the first DQ transition (0 to 1) was found for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. These parameters are read-only. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)

34.5.110 Control Register 109 (DDRMC_CR109)

Address: 400A_E000h base + 1B4h offset = 400A_E1B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															RDLVL_OFF_DIR1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	RDLVL_OFF_DL1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR109 field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 RDLVL_OFF_ DIR1	Specifies the direction of the offset for the read level midpoint delay for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. • 0 = Subtract delay • 1 = Add delay This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)
15–0 RDLVL_OFF_ DL1	Offset for the read level midpoint delay for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)

34.5.111 Control Register 110 (DDRMC_CR110)

Address: 400A_E000h base + 1B8h offset = 400A_E1B8h

Bit	31–24	23–16	15–8	7–0
R	0	RDLVL_GTDL_1	0	RDLVL_DL_1
W				
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

DDRMC_CR110 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 RDLVL_GTDL_1	Number of delay elements where gate is aligned to rising edge of DQS for data slice 1. This is readonly if the RDLVL_GATE_EN parameter is 1. If both the RDLVL_GATE_EN and the RDLVL_GATE_REG_EN parameters are 0, this can only be programmed during initialization or through software leveling. If the RDLVL_GATE_EN parameter is 0 and the RDLVL_GATE_REG_EN parameter is 1, this can be programmed at any time. **NOTE:** Value programmed in this field delays the strobe by (period of strobe/128)x the programmed value. For e.g, if the value programmed is 0x20h, the the strobe is delayed by period of strobe/4.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 RDLVL_DL_1	Number of delay elements where read DQS is placed within the DQ data eye for data slice 1. This is read-only if the RDLVL_EN parameter is 1. If both the RDLVL_EN and the RDLVL_REG_EN parameters are 0, this can only be programmed during initialization or through software leveling. If the RDLVL_EN parameter is 0 and the RDLVL_REG_EN parameteris 1, this can be programmed at any time.

34.5.112 Control Register 111 (DDRMC_CR111)

Address: 400A_E000h base + 1BCh offset = 400A_E1BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR111 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.113 Control Register 112 (DDRMC_CR112)

Address: 400A_E000h base + 1C0h offset = 400A_E1C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDLVL_END_DL2																RDLVL_BGN_DL2															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR112 field descriptions

Field	Description
31–16 RDLVL_END_ DL2	Indicates the value at which the first DQ transition (0 to 1) was found for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. These parameters are read-only. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)
15–0 RDLVL_BGN_ DL2	Indicates the value at which the first DQ transition (1 to 0) was found for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. These parameters are read-only. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)

34.5.114 Control Register 113 (DDRMC_CR113)

Address: 400A_E000h base + 1C4h offset = 400A_E1C4h

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="16" rowspan="2">RDLVL_OFF_DL2</td><td colspan="16">RDLVL_MP_DL2</td></tr>
<tr><td>W</td><td colspan="16"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

DDRMC_CR113 field descriptions

Field	Description
31–16 RDLVL_OFF_ DL2	Offset for the read level midpoint delay for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)
15–0 RDLVL_MP_DL2	Indicates the calculated midpoint value of the DQ delay for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class* = 0110)

34.5.115 Control Register 114 (DDRMC_CR114)

Address: 400A_E000h base + 1C8h offset = 400A_E1C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								RDLVL_GTDL_2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDLVL_GTDL_2								0							RDLVL_OFF_DIR2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR114 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–8 RDLVL_GTDL_2	Number of delay elements where read DQS is placed within the DQ data eye for data slice 2. This can only be programmed during initialization through software leveling.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 RDLVL_OFF_ DIR2	Specifies the direction of the offset for the read level midpoint delay for data slice X. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice. • 0 = Subtract delay • 1 = Add delay This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3(*dram_class*) = 0110)

34.5.116 Control Register 115 (DDRMC_CR115)

Address: 400A_E000h base + 1CCh offset = 400A_E1CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								RDLVL_GTDL_2							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR115 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 RDLVL_GTDL_2	Number of delay elements where gate is aligned to rising edge of DQS for data slice 2. This is readonly if the RDLVL_GATE_EN parameter is 1. If both the RDLVL_GATE_EN and the RDLVL_GATE_REG_EN parameters are 0, this can only be programmed during initialization or through software leveling. If the RDLVL_GATE_EN parameter is 0 and the RDLVL_GATE_REG_EN parameter is 1, this can be programmed at any time. **NOTE:** Value programmed in this field delays the strobe by (period of strobe/128)x the programmed value. For e.g, if the value programmed is 0x20h, the the strobe is delayed by period of strobe/4.

34.5.117 Control Register 116 (DDRMC_CR116)

Address: 400A_E000h base + 1D0h offset = 400A_E1D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR116 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.118 Control Register 117 (DDRMC_CR117)

Address: 400A_E000h base + 1D4h offset = 400A_E1D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														AXI0_ FITYPREG	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						AXI0_W_PRI		0						AXI0_R_PRI	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR117 field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 AXI0_FITYPREG	Clock domain relativity between AXI port 0 and memory controller core. Set to 0 for asynchronous or set to 2 for 1:2 port:core pseudo-sync.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 AXI0_W_PRI	Priority of write commands from AXI port 0.
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 AXI0_R_PRI	Priority of read commands from AXI port 0.

34.5.119 Control Register 118 (DDRMC_CR118)

Address: 400A_E000h base + 1D8h offset = 400A_E1D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						AXI1_W_PRI		0						AXI1_R_PRI	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR118 field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 AXI1_W_PRI	Priority of write commands from AXI port 1.
23–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 AXI1_R_PRI	Priority of read commands from AXI port 1.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.120 Control Register 119 (DDRMC_CR119)

Address: 400A_E000h base + 1DCh offset = 400A_E1DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							WRR_LATCTL	0						AXI_ASTB_DIS	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														AXI1_FITYPREG	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR119 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 WRR_LATCTL	Controls the weighted round-robin latency option. • 0 = Counters only count when their port has a command waiting to be processed. • 1 = Counters are always running.
23–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 AXI_ASTB_DIS	For certain types of AXI transfers, this parameter may be used to force a standard write transaction instead of a masked write with a read/modify/write sequence. Bit [0] controls AXI port 0, Bit [1] controls AXI port 1, etc. **Note:** When this parameter is set to 1, the AXI strobes will be ignored for the associated port for these types of transfers. This parameter applies to transfers with the following characteristics: • The transaction is aligned to the user word (in the controller core) at the starting and ending address. • The transaction is exactly one user word in length (64 bits). For each port: • 0 = Perform this operation as a masked write with a read/modify/write sequence and use the strobes. • 1 = Perform this operation as a standard write operation (not a read/modify/write) and ignore the strobes.
15–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 AXI1_FITYPREG	Clock domain relativity between AXI port 1 and memory controller core. Set to 0 for asynchronous or set to 2 for 1:2 port:core pseudo-sync.

34.5.121 Control Register 120 (DDRMC_CR120)

Address: 400A_E000h base + 1E0h offset = 400A_E1E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				AXI0_PRI1_RPRI				0				AXI0_PRI0_RPRI			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				WRR_ERR				0							W_RR_WSHR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR120 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 AXI0_PRI1_RPRI	Relative priority of priority 1 commands from port 0.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 AXI0_PRI0_RPRI	Relative priority of priority 0 commands from port 0.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 WRR_ERR	Shows the weighted round-robin arbitration errors/warnings. This parameter is read-only. • Bit [3] = The port ordering parameter values for paired ports is not sequential. • Bit [2] = The relative priority values for any of the ports paired through the W_RR_WSHR parameter are not identical • Bit [1] = Any of the relative priority parameters have been programmed to a zero value. • Bit [0] = The port ordering parameters do not all contain unique values.

Table continues on the next page...

DDRMC_CR120 field descriptions (continued)

Field	Description
	For each bit: • 0 = No error • 1 = Error
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 W_RR_WSHR	Indicates that the port pair is tied together in arbitration decisions in weighted round-robin arbitration. Bit [0] controls ports 0 and 1, Bit [1] controls port 2 and 3, etc. For each port pair: • 0 = The represented ports are treated independently in arbitration. • 1 = The represented ports are tied together for arbitration.

34.5.122 Control Register 121 (DDRMC_CR121)

Address: 400A_E000h base + 1E4h offset = 400A_E1E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															AXI0_P_ODR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				AXI0_PRI3_RPRI				0				AXI0_PRI2_RPRI			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR121 field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 AXI0_P_ODR	Used in weighted round-robin arbitration to modify the order than the ports are scanned when multiple commands are at the same priority level and have the same relative priorities.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 AXI0_PRI3_RPRI	Holds the relative priority of the AXI port 0 for priority 3 commands in weighted round robin arbitration.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 AXI0_PRI2_RPRI	Holds the relative priority of the AXI port 0 for priority 2 commands in weighted round robin arbitration.

34.5.123 Control Register 122 (DDRMC_CR122)

Address: 400A_E000h base + 1E8h offset = 400A_E1E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				AXI1_PRI1_RPRI				0				AXI1_PRI0_RPRI				0						AXI0_PRIRLX									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR122 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 AXI1_PRI1_RPRI	Holds the relative priority of AXI port 1 for priority 1 commands in weighted round robin arbitration.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 AXI1_PRI0_RPRI	Holds the relative priority of AXI port 1 for priority 0 commands in weighted round robin arbitration.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–0 AXI0_PRIRLX	Holds the counter value for AXI port 0 at which the priority relax condition is triggered in weighted round robin arbitration.

34.5.124 Control Register 123 (DDRMC_CR123)

Address: 400A_E000h base + 1ECh offset = 400A_E1ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															AXI1_P_ODR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				AXI1_PRI3_RPRI				0				AXI1_PRI2_RPRI			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR123 field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 AXI1_P_ODR	Used in weighted round-robin arbitration to modify the order than the ports are scanned when multiple commands are at the same priority level and have the same relative priorities.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 AXI1_PRI3_RPRI	Holds the relative priority of AXI port 1 for priority 3 commands in weighted round robin arbitration.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 AXI1_PRI2_RPRI	Holds the relative priority of AXI port 1 for priority 2 commands in weighted round robin arbitration.

34.5.125 Control Register 124 (DDRMC_CR124)

Address: 400A_E000h base + 1F0h offset = 400A_E1F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							MEM_RST_VAL	0						CKE_STAT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						AXI1_PRIRLX									
W							AXI1_PRIRLX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR124 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 MEM_RST_VAL	Register access to mem_rst_valid signal.
23–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 CKE_STAT	Register access to cke_status signal.

Table continues on the next page...

DDRMC_CR124 field descriptions (continued)

Field	Description
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–0 AXI1_PRIRLX	Holds the counter value for AXI port 1 at which the priority relax condition is triggered in weighted round robin arbitration.

34.5.126 Control Register 125 (DDRMC_CR125)

Address: 400A_E000h base + 1F4h offset = 400A_E1F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			PHY_WRLAT					DLL_RST_ADJ_DLY								DLL_RST_DELAY															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR125 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 PHY_WRLAT	Holds the calculated PHY tPHY_WRLAT timing parameter.
23–16 DLL_RST_ADJ_DLY	Specifies the minimum number of controller clocks after the master delay value is programmed before the DLL reset may be asserted.
15–0 DLL_RST_DELAY	Sets the number of controller clocks that the DLL reset must be held asserted for the DLL. If the DLL RESET signal is being used in the system, this parameter may be useful. If the DLL RESET signal is not being used, this parameter may be ignored.

34.5.127 Control Register 126 (DDRMC_CR126)

Address: 400A_E000h base + 1F8h offset = 400A_E1F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved		PHY_RDDATA_EN						0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PHY_RDLAT						0	UP_ERR_STAT						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR126 field descriptions

Field	Description
31–30 Reserved	This field is reserved.
29–24 PHY_RDDATA_EN	Holds the calculated value of the t_{rddata_en} timing parameter, the number of PHY clocks from the assertion of a read command on the DFI to the assertion of the PHY read data enable signal. This parameter is used to adjust the PHY read data enable signal timing. The minimum supported value for the rdlat_adj parameter is 2, and the minimum supported value for PHY read data enable is 1.
23–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–8 PHY_RDLAT	Holds the t_{phy_rdlat} timing parameter, the maximum number of PHY clocks allowed from the assertion of the PHY read data enable signal to the assertion of the PHY read data valid signal. **NOTE:** The reset value of this bit can change after reset based on internal logic.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 UP_ERR_STAT	Reports on errors in the PHY update process. This parameter will be valid when the PHY update error interrupt (bit 15) in the int_stat parameter is set to ’b1. If multiple errors occur, only the last error will be reported in this parameter.

34.5.128 Controla Register 127 (DDRMC_CR127)

Address: 400A_E000h base + 1FCh offset = 400A_E1FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															DRAM_CK_DI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR127 field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DRAM_CK_DI	This bit enables/disables clk to the memory 0 Enables 1 Disabled clk is gated

34.5.129 Control Register 128 (DDRMC_CR128)

Address: 400A_E000h base + 200h offset = 400A_E200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR128 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.130 Control Register 129 (DDRMC_CR129)

Address: 400A_E000h base + 204h offset = 400A_E204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR129 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.131 Control Register 130 (DDRMC_CR130)

Address: 400A_E000h base + 208h offset = 400A_E208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR130 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.132 Control Register 131 (DDRMC_CR131)

Address: 400A_E000h base + 20Ch offset = 400A_E20Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHYUPD_INT																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR131 field descriptions

Field	Description
31–0 PHYUPD_INT	Holds the PHY $t_{CTRLUPD\_INTERVAL}$ timing parameter. If programmed to a non-zero, a timing violation will cause an interrupt and bit (0) set in the UP_ERR_STAT parameter. Holds the PHY t timing parameter, the maximum number of PHY clocks that the MC may wait between assertions of the PHY controller update request. $_{ctrlupd\_interval}$

34.5.133 Control Register 132 (DDRMC_CR132)

Address: 400A_E000h base + 210h offset = 400A_E210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																			WRLAT_ADJ					0		RDLAT_ADJ					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR132 field descriptions

Field	Description
31–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 WRLAT_ADJ	Adjusts the relative timing in memory clocks between PHY write commands and the PHY write data enable signal to conform to PHY timing requirements. **NOTE:** For LPDDR2 memory systems (dram_class = ’b0101), the wrlat_adj value is incremented inside the memory controller by 1 to compensate for the tDSS value from the memory specification.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR132 field descriptions (continued)

Field	Description
5–0 RDLAT_ADJ	Adjusts the relative timing between PHY read commands and the PHY read data enable signal to conform to PHY timing requirements. Setting this parameter to 0x2 will cause the PHY read data enable signal will assert one controller clock after the PHY address signal. Each incremental increase will further delay the PHY read data enable signal. The minimum supported value for this parameter is 2.

34.5.134 Control Register 133 (DDRMC_CR133)

Address: 400A_E000h base + 214h offset = 400A_E214h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR133 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.135 Control Register 134 (DDRMC_CR134)

Address: 400A_E000h base + 218h offset = 400A_E218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR134 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.136 Control Register 135 (DDRMC_CR135)

Address: 400A_E000h base + 21Ch offset = 400A_E21Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR135 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.137 Control Register 136 (DDRMC_CR136)

Address: 400A_E000h base + 220h offset = 400A_E220h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR136 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.138 Control Register 137 (DDRMC_CR137)

Address: 400A_E000h base + 224h offset = 400A_E224h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												PHYCTL_DL				0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR137 field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 PHYCTL_DL	Holds the DFI t $_{ctrl\_delay}$ timing parameter, the number of PHY clocks after an assertion or de-assertion of the PHY control signals that the control signals at the PHY-DRAM interface will reflect the assertion or de-assertion. **NOTE:** The reset value of this bit can change after reset based on internal logic.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.139 Control Register 138 (DDRMC_CR138)

Address: 400A_E000h base + 228h offset = 400A_E228h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHY_WRLV_MXDL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					PHYDRAM_CK_EN			0				PHYDRAM_CK_DIS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR138 field descriptions

Field	Description
31–16 PHY_WRLV_ MXDL	Defines the maximum number of delay elements that the hardware write leveling logic will test for the “0” to “1” transition. This parameter is used by the hardware write leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 PHYDRAM_CK_ EN	Holds the PHY t $_{dram\_ck\_en}$ timing parameter, the number of PHY clocks from the de-assertion of the PHY DRAM clock disable signal on the PHY until the first valid rising edge of the clock to the DRAM memories, at the PHY-DRAM boundary. This parameter is used to indicate the number of clocks that the PHY requires to respond to a deassertion of the PHY DRAM clock disable.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 PHYDRAM_CK_ DIS	Holds the DFI t $_{dram\_ck\_dis}$ timing parameter, the number of PHY clocks from the assertion of the PHY DRAM clock disable signal on the DFI until the clock to the DRAM memories, at the PHY-DRAM boundary, maintains a low value.

34.5.140 Control Register 139 (DDRMC_CR139)

Address: 400A_E000h base + 22Ch offset = 400A_E22Ch

Bit	31–24	23–16	15–8	7–0
R/W	PHY_WRLV_RESPLAT	PHY_WRLV_LOAD	PHY_WRLV_DLL	PHY_WRLV_EN
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

DDRMC_CR139 field descriptions

Field	Description
31–24 PHY_WRLV_ RESPLAT	Defines the number of PHY clocks after a write level strobe until the response is valid. All PHY timing parameters must be programmed relative to the PHY clock. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
23–16 PHY_WRLV_ LOAD	Defines the minimum number of PHY clocks required after the write leveling delays are loaded until the first write leveling load may be asserted. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
15–8 PHY_WRLV_DLL	Defines the number of DFI clocks required for the write leveling delay line to update after a load. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
7–0 PHY_WRLV_EN	Defines the number of PHY clocks required for the write leveling delay line to update after a load. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)

34.5.141 Control Register 140 (DDRMC_CR140)

Address: 400A_E000h base + 230h offset = 400A_E230h

Bit	31–10	9–0
R	0	PHY_WRLV_WW
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

DDRMC_CR140 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–0 PHY_WRLV_ WW	Sets the minimum number of PHY clocks that must occur between write level strobes. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems:

Table continues on the next page...

DDRMC_CR140 field descriptions (continued)

Field	Description
	DDR3 (dram_class = ’b0110)

34.5.142 Control Register 141 (DDRMC_CR141)

Address: 400A_E000h base + 234h offset = 400A_E234h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHY_WRLV_RESP																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR141 field descriptions

Field	Description
31–0 PHY_WRLV_ RESP	Defines the maximum number of DFI clocks that may occur between a write level request and the write level mode enable. This parameter is used by the write leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)

34.5.143 Control Register 142 (DDRMC_CR142)

Address: 400A_E000h base + 238h offset = 400A_E238h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHY_WRLV_MAX																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR142 field descriptions

Field	Description
31–0 PHY_WRLV_ MAX	When write leveling is performed, this parameter defines the maximum number of PHY clocks that the memory controller will wait for a response from the PHY.

34.5.144 Control Register 143 (DDRMC_CR143)

Address: 400A_E000h base + 23Ch offset = 400A_E23Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDLV_GAT_MXDL																RDLV_MXDL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR143 field descriptions

Field	Description
31–16 RDLV_GAT_ MXDL	Defines the maximum number of delay elements that may be included in the gate delay line. This parameter is used by the hardware gate training logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
15–0 RDLV_MXDL	Defines the maximum number of delay elements that may be included in the read leveling delay line. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)

34.5.145 Control Register 144 (DDRMC_CR144)

Address: 400A_E000h base + 240h offset = 400A_E240h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHY_RDLVL_RES								PHY_RDLV_LOAD								PHY_RDLV_DLL								PHY_RDLV_EN							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR144 field descriptions

Field	Description
31–24 PHY_RDLVL_ RES	Defines the number of PHY clocks after a read command until the response is valid. All PHY timing parameters must be programmed relative to the PHY clock. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
23–16 PHY_RDLV_ LOAD	Defines the minimum number of PHY clocks required after the read leveling delays are loaded until the first read leveling load may be asserted. All PHY timing parameters must be programmed relative to the PHY clock. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
15–8 PHY_RDLV_DLL	Defines the number of PHY clocks required for the read leveling delay line to update after a load. All PHY timing parameters must be programmed relative to the PHY clock. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
7–0 PHY_RDLV_EN	Defines the minimum number of PHY locks required after the read leveling enable signal is asserted until the first read leveling load or read command may be asserted. All PHY timing parameters must be programmed relative to the PHY clock. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)

34.5.146 Control Register 145 (DDRMC_CR145)

Address: 400A_E000h base + 244h offset = 400A_E244h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						PHY_RDLV_RR									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR145 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–0 PHY_RDLV_RR	Sets the minimum number of PHY clocks that must occur between read commands. All PHY timing parameters must be programmed relative to the PHY clock. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 'b0110)

34.5.147 Control Register 146 (DDRMC_CR146)

Address: 400A_E000h base + 248h offset = 400A_E248h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHY_RDLVL_RESP																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR146 field descriptions

Field	Description
31–0 PHY_RDLVL_ RESP	Defines the maximum number of PHY clocks that may occur between a read level request and the read level mode enable. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 'b0110)

34.5.148 Control Register 147 (DDRMC_CR147)

Address: 400A_E000h base + 24Ch offset = 400A_E24Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												RDLV_RESP_MASK																			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR147 field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–0 RDLV_RESP_ MASK	Used to mask unused bits in the PHY read level response input signal if necessary. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = 'b0110)

34.5.149 Control Register 148 (DDRMC_CR148)

Address: 400A_E000h base + 250h offset = 400A_E250h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							RDLVL_EN	0				RDLV_GATE_RESP_MASK			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDLV_GATE_RESP_MASK															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR148 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 RDLVL_EN	Enables the hardware read leveling module in the memory controller. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110) 0 Disable 1 Enable
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–0 RDLV_GATE_ RESP_MASK	Used to mask unused bits in the PHY read level response input signal if necessary. This parameter is used by the hardware gate training logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)

34.5.150 Control Register 149 (DDRMC_CR149)

Address: 400A_E000h base + 254h offset = 400A_E254h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							RDLV_GT_ CHKENS	0							RDLVL_GATE_ EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR149 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 RDLV_GT_ CHKENS	Enables the preamble check sequence during the initial gate training sequence of the hardware read leveling module in the MC. 0 Disable 1 Enable
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 RDLVL_GATE_ EN	Enables the hardware gate training mode of the read leveling module in the memory controller. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110) 0 Disable 1 Enable

34.5.151 Control Register 150 (DDRMC_CR150)

Address: 400A_E000h base + 258h offset = 400A_E258h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHY_RDLVLMAX																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR150 field descriptions

Field	Description
31–0 PHY_ RDLVLMAX	When read leveling is performed, this parameter defines the maximum number of PHY clocks that the memory controller will wait for a response from the PHY.

34.5.152 Control Register 151 (DDRMC_CR151)

Address: 400A_E000h base + 25Ch offset = 400A_E25Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		RDLV_ERR_STA														0				RDLV_GAT_DQ_ZERO_CNT				0				RDLVL_DQ_ZERO_CNT			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR151 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–16 RDLV_ERR_STA	Reports on errors in the hardware read leveling process. This parameter will be valid when either the read leveling error interrupt or the gate training error interrupt (bits 12 or 13) in the int_stat parameter are set to ’b1. This parameter is read-only.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 RDLV_GAT_DQ_ ZERO_CNT	Defines the number of consecutive “0” responses that the controller must receive in the hardware gate training process to consider a transition from “1” to “0” to be valid. This parameter is used by the hardware gate training logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 RDLVL_DQ_ ZERO_CNT	Defines the number of consecutive “0” responses that the controller must receive in the read leveling process to consider a transition from “1” to “0” to be valid. This parameter is used by the hardware read leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)

34.5.153 Control Register 152 (DDRMC_CR152)

Address: 400A_E000h base + 260h offset = 400A_E260h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDLV_GTREFINT																RDLV_REFINT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR152 field descriptions

Field	Description
31–16 RDLV_ GTREFINT	Sets the maximum number of refreshes allowed between automatic gate training commands. Clearing this parameter to 0x0 will disable automatic gate training. This parameter is used by the hardware gate training logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)
15–0 RDLV_REFINT	Sets the maximum number of refreshes allowed between automatic read leveling operations. Clearing this parameter to 0x0 will disable automatic read leveling. This parameter is used by the leveling logic and is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110)

34.5.154 Control Register 153 (DDRMC_CR153)

Address: 400A_E000h base + 264h offset = 400A_E264h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							EN_1T_TMG	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR153 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 EN_1T_TMG	Enables the 1T timing mode in a system supporting both 1T and 2T timing. When using 1T timing in a system designed for 2T timing, the command must be modified to only assert on a single PHY clock. Setting this parameter to ’b1 takes care of this. The command will be aligned to the assertion of the chip select signal. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110) 0 Disable 1 Enable
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.155 Control Register 154 (DDRMC_CR154)

Address: 400A_E000h base + 268h offset = 400A_E268h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PAD_ZQ_EARLY_CMP_EN_TIMER					0				PAD_ZQ_MODE		0	DDR_SEL_PAD_Contr		DDR_SEL_ZQ_PAD_Contr	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	PAD_ZQ_HW_FOR	PAD_ZQ_CMP_OUT_SMP		0											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR154 field descriptions

Field	Description
31–27 PAD_ZQ_ EARLY_CMP_ EN_TIMER	This timer defines the time between the enabling of the comparator in the ZQ calibration pad and the beginning of the calibration process with the pad. 0x0 - 0x6 - Reserved 0x7 - 8 cycles .. 0x0D - 14 cycles (recommended setting) .. 0x1E - 31 cycles 0x1F - 32 cycles
26–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–21 PAD_ZQ_MODE	PAD calibration modes 0x0 No pad calibration issued (Default) 0x1 Pad calibration issued whenever memory controller issues a external memory short or Long (ZQCL or ZQCS) commands, including during memory initialization and after self refresh exit (provided controller has been configured for this). 0x2 pad calibration issued whenever memory controller drives only long calibration command to the external memory (ZQCL), including during memory initialization. 0x3 reserved
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–18 DDR_SEL_PAD_ Contr	Programmable Control bits for Other DDR-Pads, like: • DDR_SEL = 11 (DDR3 mode). • DDR_SEL = 10 (LPDDR2 mode).

Table continues on the next page...

DDRMC_CR154 field descriptions (continued)

Field	Description
17–16 DDR_SEL_ZQ_ PAD_Contr	Programmable Control bits for ZQ-Pads, like: • DDR_SEL = 00 (DDR3 mode) • DDR_SEL = 00 (LPDDR2 mode)
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 PAD_ZQ_HW_ FOR	Writing a 1 to this bit will force the hardware pad calibration once. It is the user responsibility to ensure that this bit is asserted at the correct time and when there is no traffic on the DDR interface. The HW PAD calibration staus bit in DDR_CR 156[PAD_ZQ_HW_IN_PROG] is set during the duration of the Pad calibration seq. When the H/W calibration is done, DDR_CR 156[AD_ZQ_HW_IN_PROG] will be cleared by the logic and the pad calibration values in DDR_CR 156[PAD_ZQ_HW_PU_RES] and PAD_ZQ_HW_PD_RES are valid. **NOTE:** It is not required to explicitely clear this bit before the next write. Writing a 0 in this bit has no effect on the PAD calibration logic. Even if this bit is left uncleared after last write, user needs to write a 1 again to this bit to force H/W pad calibration.
13–12 PAD_ZQ_CMP_ OUT_SMP	Defines the amount of cycles between driving the calibration signals to the ZQ calibration pad and sampling the comparator output for result. 00 7 cycles (Default) 01 15 cycles 10 23 cycles 11 31 cycles
11–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.156 Control Register 155 (DDRMC_CR155)

Address: 400A_E000h base + 26Ch offset = 400A_E26Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0				PAD_IBE1	PAD_IBE0	PAD_IBE_SEL1	PAD_IBE_SEL0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				AXI1_AWCACHE	AXI0_AWCACHE	AXI1_COBUF	AXI0_COBUF	0		PAD_ODT_BYTE1			PAD_ODT_BYTE0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR155 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 PAD_IBE1	Input buffer enable state for DDR pads corresponding to byte lane 1 (DQ[15:8], DQS[1] and DM[1]). This value takes effect only when PAD_IBE_SEL1 (bit 17) is high, otherwise the buffer enable signal is controlled by the logic.
18 PAD_IBE0	Input buffer enable state for DDR pads corresponding to byte lane 0 (DQ[7:0], DQS[0] and DM[0]). This value takes effect only when PAD_IBE_SEL0 (bit 16) is high, otherwise the buffer enable signal is controlled by the logic.
17 PAD_IBE_SEL1	DDR pad input buffer select override from software for signals corresponding to Byte lane1(DQ[15:8], DQS[1] and DM[1]). If this bit is set then DDR pad IBE is controlled by the state of PAD_IBE1 bit otherwise it is controlled by logic.
16 PAD_IBE_SEL0	DDR pad input buffer select override from software for signals corresponding to Byte lane0(DQ[7:0], DQS[0] and DM[0]).If this bit is set then DDR pad IBE is controlled by the state of PAD_IBE0 bit otherwise it is controlled by logic.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 AXI1_AWCACHE	When this bit is set, it forces the AXI port 1 AWCACHE[0] signal for the DDR controller to be set. If this bit is 0, AXI bus AWCACHE signal works as usual.
10 AXI0_AWCACHE	When this bit is set, it forces the AXI port 0 AWCACHE[0] signal for the DDR controller to be set. If this bit is 0, AXI bus AWCACHE signal works as usual.
9 AXI1_COBUF	Control for AXI1_COBUF signal. AXI port 1 coherent bufferable selection. If the AXI1_AWCACHE bit is set to ’b1 for Bufferable operation, this signal determines exactly what type of bufferable response is returned to the user interface. For guaranteed data coherency across all ports, the user should send all commands with AXI1 Coherent Bufferable signal set to ’b1. 0 Response returned when command and data have been received by the port. 1 Response returned when command accepted into the memory controller core command queue and all associated data has been received by the AXI data port.
8 AXI0_COBUF	Control for AXI0_COBUF signal. AXI port 0 coherent bufferable selection. If the AXI0_AWCACHE bit is set to ’b1 for Bufferable operation, this signal determines exactly what type of bufferable response is returned to the user interface. For guaranteed data coherency across all ports, the user should send all commands with AXI0 Coherent Bufferable signal set to ’b1. 0 Response returned when command and data have been received by the port. 1 Response returned when command accepted into the memory controller command queue and all associated data has been received by the AXI data port.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–3 PAD_ODT_ BYTE1	Termination setting for PADs related to BYTE1 (DQ,DQS and DM) in DDR3 mode only. Not applicable in LPDDR2 mode. 000 termination disabled 001 120 Ohm 010 60 Ohm

Table continues on the next page...

DDRMC_CR155 field descriptions (continued)

Field	Description
	011 40 Ohm 100 30 Ohm 101 24 Ohm 110 20 Ohm 111 17 Ohm
2–0 PAD_ODT_ BYTE0	Termination setting for PADs related to BYTE0 (DQ,DQS and DM) in DDR3 mode only. Not applicable in LPDDR2 mode.: 000 termination disabled 001 120 Ohm 010 60 Ohm 011 40 Ohm 100 30 Ohm 101 24 Ohm 110 20 Ohm 111 17 Ohm

34.5.157 Control Register 156 (DDRMC_CR156)

Address: 400A_E000h base + 270h offset = 400A_E270h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			PAD_ZQ_HW_IN_PROG	0	PAD_ZQ_HW_PD_RES					PAD_ZQ_HW_PU_RES					0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR156 field descriptions

Field	Description
31–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR156 field descriptions (continued)

Field	Description
12 PAD_ZQ_HW_ IN_PROG	Status bit when set indicates PAD H/W calibration seq in progress. When clear indicates that no H/W calibration seq is in progress.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–6 PAD_ZQ_HW_ PD_RES	Final value of the pad calibration Pull Down leg during H/W pad calibration sequence. 00000 - Max resistance ... ... 11111 - Min resistance
5–1 PAD_ZQ_HW_ PU_RES	Final value of the pad calibration PU leg during H/W pad calibration sequence. 00000 - Min resistance ... ... 11111 - Max resistance
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.158 Control Register 157 (DDRMC_CR157)

Address: 400A_E000h base + 274h offset = 400A_E274h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RD_PEND_CNT								0				RD_PEND_OFLO	RD_PEND_UFLO	PAD_IBE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR157 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR157 field descriptions (continued)

Field	Description
15–8 RD_PEND_CNT	Should be read after all txns are done on DDR. Should be clear if all reads are done.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 RD_PEND_ OFLO	Read pending counter overflow error condition. Should not occur for normal operation.
2 RD_PEND_ UFLO	Read pending counter underflow error condition. Should not occur for normal operation.
1–0 PAD_IBE	Final Input buffer value going to DDR pad. PAD_IBE[1] is for byte lane 1 signals (DQ[15:8], DQS[1] and DM[1]) PAD_IBE[0] is for byte lane 0 signals (DQ[7:0], DQS[0] and DM[0])

34.5.159 Control Register 158 (DDRMC_CR158)

Address: 400A_E000h base + 278h offset = 400A_E278h

Bit	31–6	5–0
R	0	TWR
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0

DDRMC_CR158 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 TWR	DRAM TWR value in cycles.

34.5.160 Control Register 159 (DDRMC_CR159)

Address: 400A_E000h base + 27Ch offset = 400A_E27Ch

Bit	31–16	15–0
R	0	0
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

DDRMC_CR159 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR159 field descriptions (continued)

Field	Description
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.161 Control Register 160 (DDRMC_CR160)

Address: 400A_E000h base + 280h offset = 400A_E280h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR160 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.162 Control Register 161 (DDRMC_CR161)

Address: 400A_E000h base + 284h offset = 400A_E284h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															ODT_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				TODTH_RD				0				TODTH_WR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_CR161 field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_CR161 field descriptions (continued)

Field	Description
16 ODT_EN	Enables the use of the DRAM ODT pin. This parameter is only applicable when the MC is programmed for the following memory systems: DDR3 (dram_class = ’b0110) 0 The ODT output will not be used. 1 The Controller will assert and deassert the ODT output to DRAM as needed
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 TODTH_RD	Defines the DRAM timing for the minimum ODT high time after an ODT assertion for read data. When these bits are programmed to a value larger than the number of cycles required for a burst, the command-to-command delays must be adjusted to delay commands as needed. For LPDDR2 memories: This parameter has no meaning for this memory type. For DDR3 memories: Program to the tODTH4 or the tODTH8 value from the memory specification, based on the burst length.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 TODTH_WR	Defines the DRAM timing for the minimum ODT high time after an ODT assertion for write data. When this parameter is programmed to a value larger than the number of cycles required for a burst, the command-to-command delays must be adjusted to delay commands as needed. For LPDDR2 memories: This parameter has no meaning for this memory type. For DDR3 memories: Program to the tODTH4 or the tODTH8 value from the memory specification, based on the burst length.

34.5.163 PHY Register 00 (DDRMC_PHY00)

PHY DQ timing register for data slice 0

Address: 400A_E000h base + 400h offset = 400A_E400h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSEL_START				TSEL_END				0	OE_START			0	OE_END		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY00 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DDRMC_PHY00 field descriptions (continued)

Field	Description
15–12 TSEL_START	Defines the DQ pad dynamic termination select enable time. Larger values add greater delay to when tsel turns on. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 2.
11–8 TSEL_END	Defines the DQ pad dynamic termination select disable time. Larger values increase the delay to when tsel turns off. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 2
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 OE_START	Adjusts the starting point of the DQ pad output enable window. Lower numbers pull the rising edge earlier in time, and larger numbers cause the rising edge to be delayed. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 1.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 OE_END	Adjusts the ending point of the DQ pad output enable window. Lower numbers pull the falling edge earlier in time, and larger numbers cause the falling edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. This field must be set to at least the value of bits [6:4]+2 to prevent disabling the pad before the data is completely written. Recommended setting is 0 x 3.

34.5.164 PHY Register 01 (DDRMC_PHY01)

PHY DQS timing register for data slice 0

Address: 400A_E000h base + 404h offset = 400A_E404h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TSEL_START				TSEL_END				OE_START				OE_END			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY01 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 TSEL_START	Defines the DQS pad dynamic termination select enable time. Larger values add greater delay to when tsel turns on. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 2.
11–8 TSEL_END	Defines the DQS pad dynamic termination select disable time. Larger values increase the delay to when tsel turns off. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 6.
7–4 OE_START	Adjusts the starting point of the DQS pad output enable window. Lower numbers pull the rising edge earlier in time, and larger numbers cause the rising edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. Recommended setting for this field is 0 x 1.
3–0 OE_END	Adjusts the ending point of the DQS pad output enable window. Lower numbers pull the falling edge earlier in time, and larger numbers cause the falling edge to be delayed. Each bit changes the output enable time by a 1/2 cycle resolution. This field must be set to at least the value of bits [7:4]+2 to prevent disabling the pad before the data is completely written. Recommended setting for this field is 0 x 5.

34.5.165 PHY Register 02 (DDRMC_PHY02)

PHY Gate Control Register for data slice 0.

Address: 400A_E000h base + 408h offset = 400A_E408h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		EN_HALF_CAS	0	SW_HALF_ CYCLE_SHIFT	0	WRLVL_CLKDL	0			RD_DL_SET			0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													Gate_CFG		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY02 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 EN_HALF_CAS	Subtracts 1/2 cycle from the DQS gate value programmed into PHY_GATE_CRn[2:0] by 1/2 cycle. Default 0x0. This is used when the gate is being aligned to the first DQS, and then is removed to move the gate back into the center of the preamble. 0 Do not adjust 1 Adjust the DQS gate by 1/2 clock forward.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SW_HALF_ CYCLE_SHIFT	When PHY_GB_CR[EN_SW_HALF_CYLCLE] is set to ’b1, this setting will override the calculated phase shift detection in the slice between clk and DLL-delayed version of clk_ref for the write DQS (clk_wrdqs). 0 No effect 1 Adds a half clock delay to the write level delay line.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 WRLVL_CLKDL	Adds an extra clock cycle to the delay in the PHY. 0 No change 1 Adds extra clock to the PHY if the delay is at max value, it is not already asserted, and the rising edge has not been located.

Table continues on the next page...

DDRMC_PHY02 field descriptions (continued)

Field	Description
24–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–19 RD_DL_SET	Read delay setting for io_datain FIFO timing. Recommended setting: 0 x 4
18–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 Gate_CFG	Coarse adjust of gate open time. Decreasing this value pulls the gate earlier in time. This field is of use during DQS gate training sequence.

34.5.166 PHY Register 03 (DDRMC_PHY03)

PHY DLL Master Control Register for data slice 0.

Address: 400A_E000h base + 40Ch offset = 400A_E40Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DLL_START_POINT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY03 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 DLL_START_ POINT	This value is loaded into the DLL at initialization and is the value at which the DLL will begin searching for a lock. This field must be set to a value greater than or equal to 4.

34.5.167 PHY Register 04 (DDRMC_PHY04)

PHY DLL Slave Control Register for data slice 0. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice.

Address: 400A_E000h base + 410h offset = 400A_E410h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	DLL_WRITE_DL							0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY04 field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 DLL_WRITE_DL	DLL Write delay This is delay setting for the dll line that controls the phase relationship for the write data. **NOTE:** Value programmed in this field delays the strobe by (period of strobe/128)x the programmed value. For e.g, if the value programmed is 0x20h, the the strobe is delayed by period of strobe/4.
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.168 PHY Register 10 (DDRMC_PHY10)

PHY Loopback Control Register. Reports status for the PHY for data slice 0

Address: 400A_E000h base + 428h offset = 400A_E428h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							DQS_ERROR_STATUS	LPBK_DQ_DATA							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LPBK_DQ_DATA								LPBK_DM_DATA				0		LPBK_STATUS	LPBK_START
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY10 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 DQS_ERROR_ STATUS	Status signal to indicate that the logic gate had to be forced closed. -- ’b0 = Normal operation -- ’b1 = Gate close was forced
23–8 LPBK_DQ_DATA	Reports the actual data or expected data, depending on the setting of the PHY02[RD_DL_SET] parameter bit.
7–4 LPBK_DM_DATA	Reports the actual data mask or the expected data mask, depending on the setting of the phy_gate_lpbk_ctrl_reg_fN_X [20] parameter bit.

Table continues on the next page...

DDRMC_PHY10 field descriptions (continued)

Field	Description
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 LPBK_STATUS	Reports status of loopback errors. * 'b0 = Last Loopback test had no errors. * 'b1 = Last Loopback test contained data errors.
0 LPBK_START	Defines the status of the loopback mode. * 'b0 = Not in loopback mode * 'b1 = In Loopback mode

34.5.169 PHY Register 11 (DDRMC_PHY11)

PHY DLL Status Register 0 for data slice 0.

Address: 400A_E000h base + 42Ch offset = 400A_E42Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DLL_UNLOCK_VALUE								LOCK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DLL_LOCK_VALUE								0							DLL_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY11 field descriptions

Field	Description
31–24 DLL_UNLOCK_VALUE	Reports the dll_lock_value if the DLL has become unlocked after being locked. This field will be cleared to 0x0 following a system reset. A value of 0x0 indicates that once the dll achieved lock, it has always remained locked.
23–16 LOCK	Holds the last 8 samples of the lock indicator
15–8 DLL_LOCK_VALUE	Reports the DLL encoder value from the master DLL to the slave DLL's
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DLL_LOCK	Indicates status of the DLL 0 DLL has not locked 1 DLL is locked

34.5.170 PHY Register 12 (DDRMC_PHY12)

PHY DLL Status Register 1 for data slice 0.

Address: 400A_E000h base + 430h offset = 400A_E430h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								DEC_OUT_WR							
W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								0							DLL_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY12 field descriptions

Field	Description
31–24 Reserved	This field is reserved.
23–16 DEC_OUT_WR	Holds the encoded value for the clock write delay line for this slice.
15–8 Reserved	This field is reserved.
7–1 Reserved	Holds the encoded value for the read delay line for this slice. This read-only field is reserved and always has the value 0.
0 DLL_LOCK	Indicates status of the DLL 0 DLL has not locked 1 DLL is locked

34.5.171 PHY Register 13 (DDRMC_PHY13)

PHY DLL Status Register 2 for data slice 0.

Address: 400A_E000h base + 434h offset = 400A_E434h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																DEC_OUT_WR_DQS															
W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY13 field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–0 DEC_OUT_WR_DQS	Reports the encoded value for the clock write DQS delay line

34.5.172 PHY Register 01 (DDRMC_PHY16)

Controls PHY slice 1. PHY DQ timing register. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice.

Address: 400A_E000h base + 440h offset = 400A_E440h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSEL_START				TSEL_END				0	OE_START			0	OE_END		
W	TSEL_START				TSEL_END					OE_START				OE_END		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY16 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 TSEL_START	Defines the DQ pad dynamic termination select enable time. Larger values add greater delay to when tsel turns on. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 2.

Table continues on the next page...

DDRMC_PHY16 field descriptions (continued)

Field	Description
11–8 TSEL_END	Defines the DQ pad dynamic termination select disable time. Larger values increase the delay to when tsel turns off. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 2
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 OE_START	Adjusts the starting point of the DQ pad output enable window. Lower numbers pull the rising edge earlier in time, and larger numbers cause the rising edge to be delayed. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 1.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 OE_END	Adjusts the ending point of the DQ pad output enable window. Lower numbers pull the falling edge earlier in time, and larger numbers cause the falling edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. This field must be set to at least the value of bits [6:4]+2 to prevent disabling the pad before the data is completely written. Recommended setting is 0 x 3.

34.5.173 PHY Register 17 (DDRMC_PHY17)

Controls PHY slice 1. PHY DQS timing register. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice.

Address: 400A_E000h base + 444h offset = 400A_E444h

Bit	31–16	15–12	11–8	7–4	3–0
R	0	TSEL_START	TSEL_END	OE_START	OE_END
W					
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

DDRMC_PHY17 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 TSEL_START	Defines the DQS pad dynamic termination select enable time. Larger values add greater delay to when tsel turns on. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 2.
11–8 TSEL_END	Defines the DQS pad dynamic termination select disable time. Larger values increase the delay to when tsel turns off. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 6.
7–4 OE_START	Adjusts the starting point of the DQS pad output enable window. Lower numbers pull the rising edge earlier in time, and larger numbers cause the rising edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. Recommended setting for this field is 0 x 1.
3–0 OE_END	Adjusts the ending point of the DQS pad output enable window. Lower numbers pull the falling edge earlier in time, and larger numbers cause the falling edge to be delayed. Each bit changes the output enable time by a 1/2 cycle resolution. This field must be set to at least the value of bits [7:4]+2 to prevent disabling the pad before the data is completely written. Recommended setting for this field is 0 x 5.

34.5.174 PHY Register 18 (DDRMC_PHY18)

PHY Gate Control Register for data slice 1.

Address: 400A_E000h base + 448h offset = 400A_E448h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		EN_HALF_CAS	0	SW_HALF_ CYCLE_SHIFT	0	WRLVL_CLKDL	0			RD_DL_SET			0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													Gate_CFG		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY18 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 EN_HALF_CAS	Subtracts 1/2 cycle from the DQS gate value programmed into PHY_GATE_CRn[2:0] by 1/2 cycle. Default 0x0. This is used when the gate is being aligned to the first DQS, and then is removed to move the gate back into the center of the preamble. 0 Do not adjust 1 Adjust the DQS gate by 1/2 clock forward.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SW_HALF_ CYCLE_SHIFT	When PHY_GB_CR[EN_SW_HALF_CYLCLE] is set to ’b1, this setting will override the calculated phase shift detection in the slice between clk and DLL-delayed version of clk_ref for the write DQS (clk_wrdqs). 0 No effect 1 Adds a half clock delay to the write level delay line.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 WRLVL_CLKDL	Adds an extra clock cycle to the delay in the PHY. 0 No change 1 Adds extra clock to the PHY if the delay is at max value, it is not already asserted, and the rising edge has not been located.

Table continues on the next page...

DDRMC_PHY18 field descriptions (continued)

Field	Description
24–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–19 RD_DL_SET	Read delay setting for io_datain FIFO timing. Recommended setting: 0 x 4
18–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 Gate_CFG	Coarse adjust of gate open time. Decreasing this value pulls the gate earlier in time. This field is of use during DQS gate training sequence.

34.5.175 PHY Register 19 (DDRMC_PHY19)

PHY DLL Master Control Register for data slice 1.

Address: 400A_E000h base + 44Ch offset = 400A_E44Ch

Bit	31–8	7–0
R	0	DLL_START_POINT
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

DDRMC_PHY19 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 DLL_START_ POINT	This value is loaded into the DLL at initialization and is the value at which the DLL will begin searching for a lock. This field must be set to a value greater than or equal to 4.

34.5.176 PHY Register 20 (DDRMC_PHY20)

PHY DLL Slave Control Register for data slice 1. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice

Address: 400A_E000h base + 450h offset = 400A_E450h

Bit	31–15	14–8	7–0
R	0	DLL_WRITE_DL	0
W			
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

DDRMC_PHY20 field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 DLL_WRITE_DL	DLL Write delay This is delay setting for the dll line that controls the phase relationship for the write data. **NOTE:** Value programmed in this field delays the strobe by (period of strobe/128)x the programmed value. For e.g, if the value programmed is 0x20h, the the strobe is delayed by period of strobe/4.
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.177 PHY Register 26 (DDRMC_PHY26)

PHY Loopback Control Register. Reports status for the PHY for data slice 1.

Address: 400A_E000h base + 468h offset = 400A_E468h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							DQS_ERROR_STATUS	LPBK_DQ_DATA							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LPBK_DQ_DATA								LPBK_DM_DATA				0		LPBK_STATUS	LPBK_START
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY26 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 DQS_ERROR_STATUS	Status signal to indicate that the logic gate had to be forced closed. -- ’b0 = Normal operation -- ’b1 = Gate close was forced
23–8 LPBK_DQ_DATA	Reports the actual data or expected data, depending on the setting of the PHY02[RD_DL_SET] parameter bit.
7–4 LPBK_DM_DATA	Reports the actual data mask or the expected data mask, depending on the setting of the phy_gate_lpbk_ctrl_reg_fN_X [20] parameter bit.

Table continues on the next page...

DDRMC_PHY26 field descriptions (continued)

Field	Description
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 LPBK_STATUS	Reports status of loopback errors. * ’b0 = Last Loopback test had no errors. * ’b1 = Last Loopback test contained data errors.
0 LPBK_START	Defines the status of the loopback mode. * ’b0 = Not in loopback mode * ’b1 = In Loopback mode

34.5.178 PHY Register 27 (DDRMC_PHY27)

PHY DLL Status Register for data slice 1.

Address: 400A_E000h base + 46Ch offset = 400A_E46Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLL_UNLOCK_VALUE								LOCK							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DLL_LOCK_VALUE								0							DLL_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY27 field descriptions

Field	Description
31–24 DLL_UNLOCK_ VALUE	Reports the dll_lock_value if the DLL has become unlocked after being locked. This field will be cleared to 0x0 following a system reset. A value of 0x0 indicates that once the dll achieved lock, it has always remained locked.
23–16 LOCK	Holds the last 8 samples of the lock indicator
15–8 DLL_LOCK_ VALUE	Reports the DLL encoder value from the master DLL to the slave DLL’s
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DLL_LOCK	Indicates status of the DLL 0 DLL has not locked 1 DLL is locked

34.5.179 PHY Register 28 (DDRMC_PHY28)

PHY DLL Status Register 2 for data slice 0

Address: 400A_E000h base + 470h offset = 400A_E470h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								DEC_OUT_WR							
W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								0							DLL_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY28 field descriptions

Field	Description
31–24 Reserved	This field is reserved.
23–16 DEC_OUT_WR	Holds the encoded value for the clock write delay line for this slice.
15–8 Reserved	This field is reserved.
7–1 Reserved	Holds the encoded value for the read delay line for this slice. This read-only field is reserved and always has the value 0.
0 DLL_LOCK	Indicates status of the DLL 0 DLL has not locked 1 DLL is locked

34.5.180 PHY Register 29 (DDRMC_PHY29)

PHY DLL Status Register 2 for data slice 1.

Address: 400A_E000h base + 474h offset = 400A_E474h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																DEC_OUT_WR_DQS															
W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY29 field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–0 DEC_OUT_WR_DQS	Reports the encoded value for the clock write DQS delay line

34.5.181 PHY Register 32 (DDRMC_PHY32)

PHY DQ timing register for slice 2.

Address: 400A_E000h base + 480h offset = 400A_E480h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TSEL_START				TSEL_END				OE_START				OE_END			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY32 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 TSEL_START	Defines the DQS pad dynamic termination select enable time. Larger values add greater delay to when tsel turns on. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 2.
11–8 TSEL_END	Defines the DQS pad dynamic termination select disable time. Larger values increase the delay to when tsel turns off. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 6.
7–4 OE_START	Adjusts the starting point of the DQS pad output enable window. Lower numbers pull the rising edge earlier in time, and larger numbers cause the rising edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. Recommended setting for this field is 0 x 1.
3–0 OE_END	Adjusts the ending point of the DQS pad output enable window. Lower numbers pull the falling edge earlier in time, and larger numbers cause the falling edge to be delayed. Each bit changes the output

Table continues on the next page...

DDRMC_PHY32 field descriptions (continued)

Field	Description
	enable time by a 1/2 cycle resolution. This field must be set to at least the value of bits [7:4]+2 to prevent disabling the pad before the data is completely written. Recommended setting for this field is 0 x 5.

34.5.182 PHY Register 33 (DDRMC_PHY33)

Controls PHY DQS timing register for slice 2.

Address: 400A_E000h base + 484h offset = 400A_E484h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TSEL_START				TSEL_END				0	OE_START			0	OE_END		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY33 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 TSEL_START	Defines the DQ pad dynamic termination select enable time. Larger values add greater delay to when tsel turns on. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 2.
11–8 TSEL_END	Defines the DQ pad dynamic termination select disable time. Larger values increase the delay to when tsel turns off. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 2
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 OE_START	Adjusts the starting point of the DQ pad output enable window. Lower numbers pull the rising edge earlier in time, and larger numbers cause the rising edge to be delayed. Each bit changes the output enable time by a 1/2 cycle resolution. Recommended setting is 0 x 1.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 OE_END	Adjusts the ending point of the DQ pad output enable window. Lower numbers pull the falling edge earlier in time, and larger numbers cause the falling edge to be delayed. Each bit changes the output enable time by a 1/4 cycle resolution. This field must be set to at least the value of bits [6:4]+2 to prevent disabling the pad before the data is completely written. Recommended setting is 0 x 3.

34.5.183 PHY Register 34 (DDRMC_PHY34)

PHY Gate Control Register for data slice 2.

Address: 400A_E000h base + 488h offset = 400A_E488h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		EN_HALF_CAS	0	SW_HALF_ CYCLE_SHIFT	0	WRLVL_CLKDL	0			RD_DL_SET			0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													Gate_CFG		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY34 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 EN_HALF_CAS	Subtracts 1/2 cycle from the DQS gate value programmed into PHY_GATE_CRn[2:0] by 1/2 cycle. Default 0x0. This is used when the gate is being aligned to the first DQS, and then is removed to move the gate back into the center of the preamble. 0 Do not adjust 1 Adjust the DQS gate by 1/2 clock forward.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SW_HALF_ CYCLE_SHIFT	When PHY_GB_CR[EN_SW_HALF_CYLCLE] is set to ’b1, this setting will override the calculated phase shift detection in the slice between clk and DLL-delayed version of clk_ref for the write DQS (clk_wrdqs). 0 No effect 1 Adds a half clock delay to the write level delay line.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 WRLVL_CLKDL	Adds an extra clock cycle to the delay in the PHY. 0 No change 1 Adds extra clock to the PHY if the delay is at max value, it is not already asserted, and the rising edge has not been located.

Table continues on the next page...

DDRMC_PHY34 field descriptions (continued)

Field	Description
24–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–19 RD_DL_SET	Read delay setting for io_datain FIFO timing. Recommended setting: 0 x 4
18–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 Gate_CFG	Coarse adjust of gate open time. Decreasing this value pulls the gate earlier in time. This field is of use during DQS gate training sequence.

34.5.184 PHY Register 35 (DDRMC_PHY35)

PHY DLL Master Control Register for data slice 2 .

Address: 400A_E000h base + 48Ch offset = 400A_E48Ch

Bit	31–8	7–0
R	0	DLL_START_POINT
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

DDRMC_PHY35 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 DLL_START_ POINT	This value is loaded into the DLL at initialization and is the value at which the DLL will begin searching for a lock. This field must be set to a value greater than or equal to 4.

34.5.185 PHY Register 36 (DDRMC_PHY36)

PHY DLL Slave Control Register for data slice 2. Data slices are the lowered numbered slices, and the highest number slice (slice 2) is the CA slice

Address: 400A_E000h base + 490h offset = 400A_E490h

Bit	31–15	14–8	7–0
R	0	DLL_WRITE_DL	0
W			
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

DDRMC_PHY36 field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 DLL_WRITE_DL	DLL Write delay This is delay setting for the dll line that controls the phase relationship for the write data. **NOTE:** Value programmed in this field delays the strobe by (period of strobe/128)x the programmed value. For e.g, if the value programmed is 0x20h, the the strobe is delayed by period of strobe/4.
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.186 PHY Register 42 (DDRMC_PHY42)

PHY Loopback Control Register. Reports status for the PHY for data slice 2.

Address: 400A_E000h base + 4A8h offset = 400A_E4A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							DQS_ERROR_ STATUS	LPBK_DQ_DATA							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LPBK_DQ_DATA								LPBK_DM_DATA				0		LPBK_STATUS	LPBK_START
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY42 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 DQS_ERROR_ STATUS	Status signal to indicate that the logic gate had to be forced closed. -- ’b0 = Normal operation -- ’b1 = Gate close was forced
23–8 LPBK_DQ_DATA	Reports the actual data or expected data, depending on the setting of the PHY02[RD_DL_SET] parameter bit.
7–4 LPBK_DM_DATA	Reports the actual data mask or the expected data mask, depending on the setting of the phy_gate_lpbk_ctrl_reg_fN_X [20] parameter bit.

Table continues on the next page...

DDRMC_PHY42 field descriptions (continued)

Field	Description
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 LPBK_STATUS	Reports status of loopback errors. * 'b0 = Last Loopback test had no errors. * 'b1 = Last Loopback test contained data errors.
0 LPBK_START	Defines the status of the loopback mode. * 'b0 = Not in loopback mode * 'b1 = In Loopback mode

34.5.187 PHY Register 43 (DDRMC_PHY43)

PHY DLL Status Register for data slice 2 .

Address: 400A_E000h base + 4ACh offset = 400A_E4ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	DLL_UNLOCK_VALUE								LOCK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DLL_LOCK_VALUE								0							DLL_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY43 field descriptions

Field	Description
31–24 DLL_UNLOCK_VALUE	Reports the dll_lock_value if the DLL has become unlocked after being locked. This field will be cleared to 0x0 following a system reset. A value of 0x0 indicates that once the dll achieved lock, it has always remained locked.
23–16 LOCK	Holds the last 8 samples of the lock indicator
15–8 DLL_LOCK_VALUE	Reports the DLL encoder value from the master DLL to the slave DLL's
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DLL_LOCK	Indicates status of the DLL 0 DLL has not locked 1 DLL is locked

34.5.188 PHY Register 44 (DDRMC_PHY44)

PHY DLL Status Register 2 for data slice 0

Address: 400A_E000h base + 4B0h offset = 400A_E4B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								DEC_OUT_WR							
W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								0							DLL_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY44 field descriptions

Field	Description
31–24 Reserved	This field is reserved.
23–16 DEC_OUT_WR	Holds the encoded value for the clock write delay line for this slice.
15–8 Reserved	This field is reserved.
7–1 Reserved	Holds the encoded value for the read delay line for this slice. This read-only field is reserved and always has the value 0.
0 DLL_LOCK	Indicates status of the DLL 0 DLL has not locked 1 DLL is locked

34.5.189 PHY Register 45 (DDRMC_PHY45)

PHY DLL Status Register 2 for data slice .

Address: 400A_E000h base + 4B4h offset = 400A_E4B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																DEC_OUT_WR_DQS															
W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY45 field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–0 DEC_OUT_WR_ DQS	Reports the encoded value for the clock write DQS delay line

34.5.190 PHY Register 49 (DDRMC_PHY49)

PHY DLL Slave Control LPDDR2 Register

Address: 400A_E000h base + 4C4h offset = 400A_E4C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PHY_RDLV_DL								0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY49 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 PHY_RDLV_DL	PHY Read Level Delay This is delay setting for the DLL line that controls the phase relationship for address and control data slice in LPDDR2 mode only. **NOTE:** This field is valid only MC is in LPDDR2 mode.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.191 PHY Register 50 (DDRMC_PHY50)

Address: 400A_E000h base + 4C8h offset = 400A_E4C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		DFI_MOBILE_EN	DDR3_MODE	DDR_SEL	0		EN_SW_HALF_CYCLE	0			CLEAR_FIFO	0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY50 field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 PHY logic configured for standard DDR operation 1 PHY gating logic configured for Mobile (LPDDR2) gating operation. The DRAM DQS signals are assumed to be pulled down on the board.
13 DFI_MOBILE_EN	DFI control for Mobile PHYs. 0 PHY gating logic configured for standard DDR 1 PHY gating logic configured for Mobile (LPDDR2) operating logic. The DRAM DQS signals are assumed to be pulled down on the board.
12 DDR3_MODE	Enables the generation of the additional DQS pulse required for DDR3 controllers. 0 No Action 1 Generate Pulse (DDR3 memories)
11 DDR_SEL	Controls the PHY memory system mode. 0 non-LPDDR2 mode 1 LPDDR2 mode
10–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 EN_SW_HALF_CYCLE	Enables the Software Half Cycle Shift 0 Hardware automatically controls any shifting needed for the write level delay line. 1 The setting in PHY_GATE_CR[SW_HALF_CYCLE_SHIFT] bit defines the shift.

Table continues on the next page...

DDRMC_PHY50 field descriptions (continued)

Field	Description
	NOTE: If the user chooses to control the half cycle shift manually, it is important that PHY_GATE_CR[SW_HALF_CYCLE_SHIFT] be cleared to 'b0 if the delay is less than a 1/2 cycle and set to 'b1 if the delay is greater than a 1/2 cycle. It is recommended to allow the hardware to control this automatically.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 CLEAR_FIFO	Clear the gather FIFO. This is an emergency reset for the pointers on the data gather FIFO. 0 No action 1 Clear the FIFO
3–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

34.5.192 PHY Register 52 (DDRMC_PHY52)

Address: 400A_E000h base + 4D0h offset = 400A_E4D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved											tsel_off_value_data	Reserved			tsel_rd_value_data
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved			tsel_off_value_dqs	Reserved			tsel_rd_value_dqs	Reserved			tsel_off_value_dm	Reserved			tsel_rd_value_dm
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DDRMC_PHY52 field descriptions

Field	Description
31–21 Reserved	This field is reserved.
20 tsel_off_value_data	Termination select off value for the data
19–17 Reserved	This field is reserved.

Table continues on the next page...

DDRMC_PHY52 field descriptions (continued)

Field	Description
16 tsel_rd_value_ data	Termination select read value for the data
15–13 Reserved	This field is reserved.
12 tsel_off_value_ dqs	Termination select off value for the data strobe
11–9 Reserved	This field is reserved.
8 tsel_rd_value_ dqs	Termination select read value for the data strobe
7–5 Reserved	This field is reserved.
4 tsel_off_value_ dm	Termination select off value for the data mask
3–1 Reserved	This field is reserved.
0 tsel_rd_value_dm	Termination select read value for the data mask

34.6 Functional Description

This section discusses the programming model and operation of the memory controller.

34.6.1 Address Mapping

The memory controller automatically maps user addresses to the DRAM memory in a contiguous block. Addressing starts at user address 0 and ends at the highest available address according to the size and number of DRAM devices present. This mapping is dependent on how the memory controller is configured.

The mapping of the address space to the internal data storage structure of the DRAM devices is based on the actual size of the DRAM devices available. The size is stored in user-programmable parameters that must be initialized at power up. Certain DRAM devices allow for different mapping options to be chosen, while other DRAM devices depend on the chosen burst length.

34.6.1.1 DDR SDRAM Address Mapping Options

The address structure of DDR SDRAM devices contains the following five fields:

- Chip select
- Row
- Bank
- Column
- Datapath

Each of these fields can be individually addressed when accessing the DRAM.

The maximum widths of the fields are based on the configuration settings. The actual widths of the fields may be smaller if the device address width parameters (DDR_CR73[ROW_DIFF], DDR_CR73[BANK_DIFF], and DDR_CR73[COL_DIFF]) are programmed differently.

34.6.1.2 Maximum Address Space

The maximum user address range is determined by the width of the memory datapath, the number of chip select pins, and the address space of the DRAM device. The maximum amount of memory can be calculated by the following formula:

$$\text{Maximum Memory Size} = \text{Chip Selects} \times 2^{\text{Address}} \times \text{Banks} \times \text{DatapathWidth}$$

See the chip configuration section of the reference manual for the maximum address space calculated for this device.

34.6.1.3 Memory Mapping to Address Space

The *col_diff* and *row_diff* parameters can each range from the maximum configured for the memory controller to seven bits smaller than the maximum configured. This allows the Memory Controller to function with a wide variety of memory sizes. The settings for the *col_diff* and *row_diff* parameters control how the address map is used to decode the

user address to the DRAM chip selects and row and column addresses. The *bank_diff* parameter controls the DRAM bank address information. It is assumed that the values in these parameters never exceed the maximum values configured.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Chip select	Row																Bank					Column									Data path

34.6.2 AXI Interface

This Memory Controller contains 2 internal AXI ports which communicate on the AXI bus. Each AXI port uses an AXI interface block to connect to the MC core. An arbitration scheme is required to control the traffic between the buses and the core.

34.6.2.1 Architecture Overview

The Multi-Port Memory Controller was designed for high memory bandwidth utilization and efficient arbitration for high priority requests.

Multi- Port system consists of the following pieces:

- 2 AXI Interfaces
- Arbiter
- Command Queue with Placement Logic
- Write Data Latency Queue
- Read Data Queues
- DRAM Command Processing
- Register Port

34.6.2.2 AXI Interfaces

The core interfaces with 2 AXI data port blocks and 1 register port. The AXI data ports function as AXI slaves to external AXI masters such as CPUs, DMAs, DSPs, and other peripherals. Transfers are burst-based of variable byte counts. The transfer types INCR and WRAP are fully supported. The AXI ports may accept commands from any AXI bus master. The AXI ID signals identify which of the 8192 AXI IDs is associated with the command. This AXI ID is concatenated with an originating port identifier to create a source ID which is used in the core to maintain originator information. The uppermost bit/s of the source ID represent the port number. There are no restrictions on mapping of AXI IDs to AXI bus masters.The AXI interfaces handle all communication between the AXI and the core. Each interface contains five separate channels of traffic to/from the AXI bus: write response, read command, write command, read data and write data. Each port will always support full-size transfers where the full data port width is utilized on each beat. In addition, each port can be independently programmed to support narrow transfers, where not all the bits of the data port width are used. This translates to a bytes-per-beat size that is less than the port data width. For a multiport configuration where the port data widths are not uniform, the transfer size that defines a narrow transfer will vary based on each port's data width. There are no fixed timing requirements on a port between the traffic channels when the narrow transfer option is disabled, and in this case, write data may arrive before, with, or after the write command. When the narrow transfer option is enabled, a port will not accept write data until it has received the command and is aware of the total byte count associated with that command.

34.6.2.3 Restrictions on the AXI Bus

For this Memory Controller, the following restrictions exist:

- FIXED burst types are not supported.
- The response signals will never respond with a DECERR response type.
- Cacheable, read allocate or write allocate commands are not supported.
- WRAP commands must be aligned.
- Write data can not be interleaved. Write data must be presented to the memory controller in the same order as the write commands.
- Each port may only issue a fixed number of narrow transfers at a time. The implementation requires that each requestor must be individually enabled for narrow transfers, and that all commands from these requestors be counted in this limit, whether they are narrow transfers or not.

- Each ports may only monitor the exclusivity of a limited number of transactions at any time. In addition, each port will only maintain one exclusive monitor per requestor at a time.
- Locked access is not configured for this memory controller

34.6.2.4 Internal Command Handling

The Memory Controller uses placement logic to fill the command queue with a command order that maximizes the throughput and efficiency of the core. Re-ordering of commands depends on the restrictions of the AXI bus. If the placement logic is being used, the Memory Controller will optimize the core by re-ordering read and write commands as necessary. Read and write commands from individual ports are not affected by other ports. The following re-ordering rules apply to any one port on this memory controller:

- Two read commands from the same AXI ID on a port will not be re-ordered and will always execute in the same order as they were accepted into their port.
- Two read commands from different AXI IDs on a port may be automatically re-ordered in the core to execute out-of-order.
  - When commands from different AXI IDs are re-ordered, read data returned to the AXI port interfaces will also be out-of-order and may be interleaved. To avoid re-ordering within a port, the AXI bus master should use one AXI ID for all commands from any port.
- Two write commands from the same AXI ID on a port must remain sequential and will not be reordered. These transactions will always occur in the order that the commands were received at the port.
- Two write commands from different AXI IDs on a port must remain sequential and will not be re-ordered. These transactions will always occur in the order that the commands were received at the port.
- A read command that follows a write command from the same AXI ID may be re-ordered to execute in an optimal order, as long as there are no collisions between the commands.
- A read command that follows a write command from a different AXI ID on a port may be reordered to execute in an optimal order, as long as there are no collisions between the commands.

- A write command that follows a read command from the same AXI ID may be re-ordered to execute in an optimal order, as long as there are no collisions between the commands.
- A write command that follows a read command from a different AXI ID on a port may be reordered to execute in an optimal order, as long as there are no collisions between the commands.

Command handling is summarized in the following table.

Table 34-199. Re-Ordering / Interleaving Behavior

Commands	AXI IDs	Can Commands be Re-Arranged and Data be Interleaved?
Two Read Commands from 1 port	Same	No
Two Read Commands from 1 port	Different	Yes
Two Read Commands from Different Ports	Same or Different	Yes
Two Write Commands from 1 port	Same	No
Two Write Commands from 1 port	Different	No
Two Write Commands from Different Ports	Same or Different	Yes
Read following a Write from 1 port	Same	Yes
Read following a Write from 1 port	Different	Yes
Read following a Write from different ports	Same or Different	Yes
Write following a Read from 1 port	Same	Yes
Write following a Read from 1 port	Different	Yes
Write following a Read from different ports	Same or Different	Yes

An incoming AXI transaction is mapped into a core-level transaction, then synchronized from the AXI clock domain to the core clock domain and stored in the AXI port FIFOs. Each instruction consists of an address, size, length and AXI ID. Since a port may utilize multiple AXI IDs, the source ID that is used in the core is a combination of both the port and identifier information. This concatenation occurs in the Arbiter and this source ID is used in the placement logic. From the AXI FIFOs, the transaction is presented to the Arbiter which arbitrates requests from all ports and forwards a single transaction to the core.

34.6.2.5 Controller configuration

- **Datapath Width** - Each port has a data interface width of 64 bits.
- **Width of the ID** - Each port is configured with a AXI ID of 13 bits.

- **Priority Definition** - Command priority is defined based on the port and the command type. For each port Y, there is an AXIY_R_PRI bit which defines priorities for all read commands and an AXIY_W_PRI bit which defines priorities for all write commands. Supported priority values range from 0 to 3, with 0 as the highest priority.
- 32 bit wide asynchronous register interface
- **Buffering** - Each data port contains a command, a read and a write FIFO, and a response storage array. In addition, each programmable port contains an asynchronous response FIFO to synchronize the memory response to the port time domain when operating asynchronously.
- **Narrow Transfer FIFO** - Since any command could be a full-size or narrow transwer, an entry is created in the port narrow transfer FIFO with the address, bytes-per-beat and a narrow transfer flag for each command. Each port has been configured with a set number of entries in this FIFO. Therefore, if a port receives enough commands to fill the narrow transfer FIFO, the associated port command FIFO may be stalled from accepting more commands. Therefore, the user should set the depths of the narrow transfer FIFOs accordingly.
- **Exclusive Access Buffer Depth** - This feature allows individual sources to monitor activity on memory locations. This type of access will only be used if exclusive access commands are issued to the memory controller by driving the AXI port Y atomic access indicator to 'b01 with a read command. Each port of this Memory Controller contains 1 exclusive buffer and therefore each port may monitor the exclusivity of up to 1 transaction at any time. Refer to Exclusive Access Option for more information.
- **Error Detection** When an illegal operational condition is detected on a new AXI transaction entering the port, the port responds through an AXI error signal, by asserting the interrupt signal from the memory controller, and recording the error signature in the register space.

  The AXI error signal flagged is dependent on the type of transaction that caused the error (read or write). If the error was associated with the command, the port command error interrupt (bit 7) will be set in the DDR_CR80[INT_STAT], and the address, source ID and type related to the error will be saved in the DDR_CR85[P_CMDERRADD], DDR_CR86[P_CMDERRID] and DDR_CR86[P_CMDERRTYP] bits.

34.6.2.6 Port Clocking

There are two user-selectable modes of operation for each of the AXI port interfaces. The mode is set by programming the corresponding *axiY_fitypreg* parameter (where Y represents the port number). The two settings are:

- **Asynchronous ('b00)**

  The AXI interface port and the core operate on clocks that are mismatched in frequency and phase. The AXI interface port FIFOs use two stages of synchronization logic to synchronize commands, write data and read data to the appropriate clock domain.

- **1:2 Port: Core Pseudo-Synchronous ('b10)**

  The AXI interface port operates at half of the frequency of the core frequency, with clocks that are aligned in phase. One stage of the two-stage synchronization logic of the FIFOs will be utilized to synchronize commands, write data and read data to the appropriate clock domain.

34.6.2.7 AXI Port FIFOs

Incoming transactions from the AXI interfaces are processed by the interface logic and mapped into equivalent transactions on the core bus. These transactions are queued into each port's command FIFO.

34.6.2.8 Command FIFO

Prior to entering the port command FIFO, the AXI port interface blocks convert the AXI transactions into core transactions.

34.6.2.8.1 In-Port Arbitration

Within each port, it is possible that both the read and write command channel of the AXI bus are active concurrently. If this occurs, each port performs a simple arbitration to select the command to pass through to the Arbiter. This arbitration is based on the following factors, in order of importance:

1. Priority of read commands versus priority of write commands for this port (values in the bits *axiY_r_pri* versus *axiY_w_pri* (where Y represents the port number)).

2. Default Read over Write preference. If both read and write commands are accepted simultaneously, and their priorities match, then the read channel will be selected first.

The following table shows the system behavior when the port FIFO is full, allowing commands to accumulate on both channels. The commands will be arbitrated into the port command FIFO based on the order that they are accepted at the port.

Table 34-200. Port Arbitration Example 1

Cycle	Port Command FIFO	Read Channel		Write Channel		Arbitrated into Port
		Command Accepted	Priority	Command Accepted	Priority	
1	FULL	1st Read	Any			None (FIFO full)
2	FULL			1st Write	Any	None (FIFO full)
3	Available					1st Read (Arrived First)
4	Available	2nd Read	Any			1st Write (Arrived First)
5	Available					2nd Read

The following table shows the system behavior when read and write commands are accepted simultaneously. The commands will be arbitrated into the port command FIFO based on priority for that type of command. When the commands are not simultaneous, they will be arbitrated based on the order that they are accepted at the port.

Table 34-201. Port Arbitration Example 2

Cycle	Port Command FIFO	Read Channel		Write Channel		Arbitrated into Port
		Command Accepted	Priority	Command Accepted	Priority	
1	Available	1st Read	Lower	1st Write	Higher	1st Write (Higher Priority)
2	Available			2nd Write	Higher	1st Read (Arrived First)
3	Available					2nd Write
4	Available	2nd Read	Lower	3rd Write	Higher	3rd Write (Higher Priority)
5	Available					2nd Read

The following table shows the system behavior when read and write commands are accepted simultaneously and their port priorities are the same. Read commands will automatically be arbitrated ahead of write commands when both other conditions match. When the commands are not simultaneous, they will be arbitrated based on the order that they are accepted at the port.

Table 34-202. Port Arbitration Example 3

Cycle	Port Command FIFO	Read Channel		Write Channel		Arbitrated into Port
		Command Accepted	Priority	Command Accepted	Priority	
1	Available	1st Read	Same	1st Write	Same	1st Read (Read over Write Priority)
2	Available	2nd Read	Same			1st Write (Arrived First)
3	Available			2nd Write	Same	2nd Read (Arrived First)
4	Available					2nd Write
5	Available	3rd Read	Same	3rd Write	Same	3rd Read (Read over Write Priority)
6	Available					3rd Write

34.6.2.9 Read FIFO

There is only one streaming read data interface out from the core for all AXI ports, regardless of the number of ports or the number of AXI IDs for any port. The Memory Controller maps this data stream back to the proper port. The AXI bus master must map the data back to the originator associated with each AXI ID.

With this singular data interface, the AXI port must be ready to accept the read data as soon as it is available on the internal core bus to avoid stalling the memory controller. The AXI port Y read data valid indicator output signal will be asserted whenever the port has received valid data from memory that can be passed back to the originator associated with that AXI ID.

If the core stalls while waiting for the read data in the read data queue to be off-loaded, the performance across all the ports in the system will be affected.

34.6.2.10 Write FIFO

The depth of the write FIFO depends on the typical byte count of a burst write transaction. Note that there is a write data queue inside the core as well. Therefore, the purpose of the write FIFO is to allow the AXI bus to off load its write data completely before the data is transferred to the core buffers. Each port has a distinct write channel into the core write data queue. If there are multiple AXI IDs for a port, they will all share the channel for that port.

While an AXI write may begin at any address, aligned or un-aligned, the protocol specifies that it must complete at an address that is aligned to the beat size.

Depending on the type of transfer, the write data may arrive before the command. However, if the originator associated with the AXI ID sending the data is capable of sending narrow transfers, then the write data will not be accepted until the command has been received.

34.6.2.11 Response Interface

When a write request is accepted into the AXI interface, an entry will be created in the Response Storage Array for that command. The array will maintain information on the order of received commands from each AXI ID for each port. Write responses will be returned in the same order as the commands were received, regardless of the AXI ID or the type of response requested (bufferable, coherent bufferable or non-bufferable). When a write response is ready, the array will verify that this is oldest command for that port and if so, the response will be sent out. The timing of this response is dependent on the type of response requested and the contents of the command queue. Responses for bufferable commands will be available when the port has received all of the data, responses for coherent bufferable commands will be available when the core has received the command and the port has received all the data and responses for non-bufferable commands will be available when all data has been sent to the DFI. It is possible that the response for a newer command for that port will be ready and waiting until the responses for all older commands for that port are ready and sent to the user interface. Write responses will be returned to the AXI master through the signal AXI port Y write response signal and its associated valid indicator, the AXI port Y read data valid indicator signal.

34.6.2.12 Bufferable, Coherent Bufferable and Non-Bufferable Response Types

DDR_CR155[AXI0_COBUF] bit controls the timing of the response being sent by the controller for a bufferable write command

When AXI0_COBUF bit is set to '0', a bufferable write command is issued, then the response for that write command is sent on the crossbar interface when the command and data reception is completed on the AXI0 port. There is no guarantee of data coherency across all AXI ports. If data coherency is NOT required across ports, this setting will provide the fastest response time for bufferable write requests.

When DDR_CR155[AXI1_COBUF] set to '1', a coherent bufferable write command is issued, then the response for that write command is sent on the crossbar interface when the command queue and data reception is completed on theAXI1 port. This guarantees data coherency across all ports but reduces the overall write response latency relative to the non-bufferable option.

34.6.2.13 Exclusive Access Option

The exclusive access feature allows a master to monitor if a memory area has been altered since its last read. Exclusive access does not imply that the memory area is locked; other AXI IDs of that port, or other ports, may access the area for reads or writes even though an exclusive access exists. If any writes occur to a memory area with a valid exclusive access request, the master will lose exclusivity and be informed of this status when it attempts to write to the area again. A loss of exclusivity does not trigger an interrupt or any error conditions; however, the AXI protocol requires that the write data is not written to memory if an exclusive write fails its exclusivity check. The master that has lost exclusivity must determine whether to restart the sequence by requesting another exclusive read or to write the data to the memory regardless via a non-exclusive write.

34.6.2.14 Error Responses

Refer to register description section for error responses and handling.

34.6.2.15 DRAM Command Processing

The DRAM command processing logic is used to process the commands in the command queue. The logic organizes the commands to the memories in such a way that data throughput is maximized. Bank opening and closing cycles are used for data transfers.

34.6.3 Multi-Port Arbiter

The Arbiter is responsible for arbitrating requests from the ports and sending requests to the MC core. Each transaction received at the Arbiter logic has an associated priority, which works with each port's arbitration logic to determine how ports issue requests to the MC core. This Memory Controller supports the Weighted Round-Robin arbitration scheme. The Arbiter logic routes read data from the MC core to the appropriate port. The requesting port is assumed able to receive the data. Write data from each port is connected directly to its own write data interface in the MC core, allowing the ports to independently pass write data to the MC core buffers.

34.6.3.1 Arbitration Overview

The weighted round-robin arbitration scheme is a three-step arbitration system. All commands are routed into priority groups based on the priority of the requests. Then, within each priority group, requests are serviced according to the "weight" (relative priority) of each port. Finally, each priority group presents a single command to the priority select module, which passes the highest priority command on to the MC core. This arbitration scheme also supports two additional features. For situations where the priority and the relative priority for multiple commands are identical, a port ordering system in included whereby the user may adjust the order in which the ports are considered. Secondly, for situations where two ports may be related, a mechanism is included which allows a pair of ports to share arbitration bandwidth for bandwidth efficiency. Weighted round-robin arbitration is a complex arbitration scheme. To understand the operation, each concept must be first understood individually. Understanding Round-Robin Operation through Understanding Port Ordering describe the various components of weighted round-robin arbitration. Note that the examples may utilize a greater number of ports and a larger number of priority levels than are available in this Memory Controller. This is done intentionally for explanation.

34.6.3.2 Understanding Round-Robin Operation

Round-robin operation is the simplest form of arbitration and is ideal for systems that do not require requests to be treated preferentially to maintain bandwidth or minimize latency. This scheme uses a counter that rotates through the port numbers, incrementing every time a port request is granted. If the port that the counter is referencing has an active request, and the MC core command queue is not full, then this request will be sent to the MC core. If there is not an active request for that port, then the port will be skipped

and the next port will be checked. The counter will increment by one whenever any request has been processed, regardless of which port's request was arbitrated. Round-robin arbitration ensures that each port's requests can be successfully arbitrated into the MC core every N cycles, where N is the number of ports in the Memory Controller. No port will ever be locked out, and any port can have its requests serviced on every cycle as long as all other ports are quiet and the command queue is not full. An example of the round-robin scheme is shown in the following table. Cycles 0, 2 and 6 show the system behavior when the command queue is full. Cycle 8 and 11 show the system behavior when the port addressed by the arbitration counter does not have an active request. In particular, note cycle 11: The port addressed by the arbitration counter (0) is not requesting, so the counter scans through the other ports, in incrementing order, to find an active request. Port 2 is requesting and therefore wins arbitration, but the counter only increments to port 1 which was the next port in the sequence. All other cycles show normal behavior.

Table 34-203. Round-Robin Operation Example

Cycle	Port Addressed by the Arbitration Counter	Ports Requesting				Command Queue Full?	Arbitration Winner	Value of Counter at Next Cycle
		P0	P1	P2	P3			
0	0	Y	Y	Y	Y	Yes	None	0
1	0	Y	Y	Y	Y	No	P0	1
2	1		Y	Y	Y	Yes	None	1
3	1	Y	Y	Y	Y	No	P1	2
4	2	Y		Y	Y	No	P2	3
5	3	Y			Y	No	P3	0
6	0	Y		Y		Yes	None	0
7	0	Y		Y		No	P0	1
8	1			Y		No	P2	2
9	2			Y	Y	No	P2	3
10	3	Y			Y	No	P3	0
11	0			Y		No	P2	1

34.6.3.3 Understanding Port Priority

For AXI ports, the priority is associated with a port and each port has separate priority parameter for reads and writes. These values are stored into the programmable bits *axiY_r_pri* and *axiY_w_pri* (where Y represents the port number) at controller initialization. Internally, the ports are organized into priority groups based on their priority setting. The priority value is also used by the placement logic inside the MC core when filling the command queue. A priority value of 0 is highest priority, and a priority

value of (decimal) 3 is the lowest priority in the Memory Controller. The user may program at priority level 0; however, it is best to reserve this priority value so that the placement queue can elevate to this level through aging.

34.6.3.4 Understanding Relative Priority

Inside each priority group, the relative priority is used to determine arbitration. The Memory Controller contains 4 identical priority groups with logic that selects between the requests from all commands at that priority level. The relative priority parameters *axiY_priZ_rpri* (where Y is the port number and Z is the priority group) "weight" the ports for each level and determine how the priority group will be arbitrated. The following figure shows this type of arbitration system.

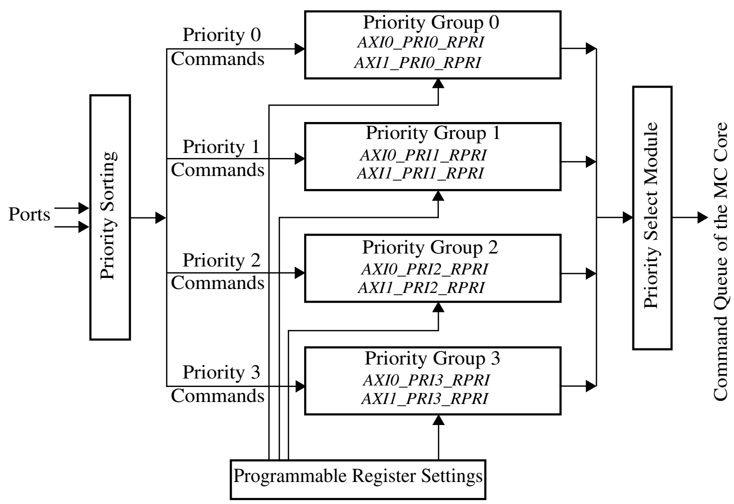


Figure 34-194. Weighted Round-Robin Priority Group Structure

By using the relative priority concept, the arbitration is skewed in favor of certain ports based on user programming. Note that the relative priority bits have a minimum acceptable value of 1 to prevent port lockout. A '0' value will cause an error condition.

If the relative priorities are all programmed to the same value within any priority group, then the arbitration will mimic a version of simple round-robin scheme within that priority group. Instead of incrementing whenever any request is processed, the simple round-robin counter will only increment to the next port after the value in the *axiY_priZ_rpri* bits number of requests are processed.

Each port Y for priority level Z will be allocated the ratio of that port's relative priority bits (*axiY_priZ_rpri*) to the sum of all requesting port's relative priority values. If a particular port is not requesting, then it is not included in the sum calculation, which means that the arbitration will be split with relative proportions among the requesting ports.

As an example, consider a system with 4 ports where all requests are at priority 0. This system is described in the following table.

Table 34-204. Relative Priority Example

Memory Controller bits	System A
AXI0_PRI0_RPRI	1
AXI1_PRI0_RPRI	2
AXI2_PRI0_RPRI	3
AXI3_PRI0_RPRI	4

For this system, port 0 will be serviced 1/(1+2+3+4) = 1/10 of the time and Port 3 will be serviced 4/ (1+2+3+4) = 4/10 of the time. However, if Port 2 is not actively requesting, then port 0 will be serviced 1/(1+2+4) = 1/7 of the time and port 3 will be serviced 4/ (1+2+4) = 4/7 of the time.

In order to ensure that relative priorities are maintained, there is a weight counter for each port within each priority group. These counters track the number of transactions accepted for that port in that priority group. When any counter value reaches the programmed relative port priority, the scan order for that priority group will be internally modified. The port that has met its relative priority will be dynamically positioned to the bottom of the scan order (and its counter will be reset), allowing other ports a preferential position.

For ports that are not expected to issue requests at a certain priority level, the associated relative priority parameter should be programmed to 0x1. This allows for minimum allocation without the risk of lock out in case a command appears.

34.6.3.5 Understanding Port Ordering

With simple round-robin arbitration, the ports are scanned based on their port number in incrementing order in the system. Assuming that the command queue is not full, the port referenced by the counter is examined for valid incoming transactions. If there is an active request, it will be accepted. Otherwise, the next port in the scan order will be checked, and its request accepted. For the Memory Controller with weighted round-robin arbitration, the user has the option of adjusting the order that the ports are scanned. This is useful if requests from certain ports are more critical, or if a specific order may reduce contention between ports. The *axiY_p_odr* bits are used to set this new scan order. A value of 'b0 gives the highest listing in the scan order, and a value of 'b1 is the lowest listing in the scan order. If the 2 *axiY_p_odr* bits are programmed with unique values, then the scan order will be modified to proceed sequentially in this new order. If any of the port ordering parameters have the same value, then those ports will still be equal in the arbitration test. In this case, the port number will select between these ports, with the lower-numbered port automatically being selected first. To demonstrate this concept,

consider a system with 8 ports and the two port orders as shown in the following table. For System B, the port ordering bits all contain unique values, so the resulting order is entirely based on the values of the parameters. For System C, three ports have the same programmed values for the port order. For these three ports, the port number sets the order. Remaining ports follow the port ordering parameters. If all of the port ordering bits are programmed with the same value, then the scan order will default to the numbered port order.

Table 34-205. Port Ordering Example

Memory Controller Parameter	System B	System C
axi0_p_odr	3	3
axi1_p_odr	4	0
axi2_p_odr	5	5
axi3_p_odr	6	6
axi4_p_odr	7	7
axi5_p_odr	0	1
axi6_p_odr	2	0
axi7_p_odr	1	0
Port Scan Order	P5-P7-P6-P0-P1-P2-P3-P4	P1-P6-P7-P5-P0-P2-P3-P4

34.6.3.6 Weighted Round-Robin Arbitration Summary

The MC weighted round-robin arbitration system combines the concepts of round-robin operation, priority, relative priority and port ordering. The incoming commands are separated into priority groups based on the priority of the associated port for that type of command. Within each priority group, the relative priority values are examined to determine the arbitration winner. If the relative priority values are identical and no individual command can be selected, then the scan order is used to select between the requests. In the end, the highest priority command, from the highest relative priority port, with the highest location in the scan order will be selected and sent to the MC core. As an example, consider the system described in Table 34-206. The counters refer to the counters that exist for each port within each priority group to ensure that relative priorities are maintained. For simplification, the command queue is considered to never be full and commands are only received at priority level 0. The behavior is shown in Table 34-207. The highest port in the scan order that is requesting always wins arbitration, and the scan order is dynamically modified when any port counter reaches its allocated relative priority value. Note that if the command queue was considered, then cycles where the command queue was full would not have any arbitration winner and therefore, the counter values and scan order would not change on that cycle.

Table 34-206. System D Specification

Memory Controller Parameter	Port 0	Port 1	Port 2	Port 3
AXIY_PRI0_RPRI	4	3	2	1
AXIY_P_ODR	0	1	2	3

Table 34-207. System D Operation

Cycle	Ports Requesting				Arbitration Winner	Next Counter				Next Scan Order
	P0	P1	P2	P3		P0	P1	P2	P3	
										P0-P1-P2-P3
0	Y			Y	P0	1	0	0	0	P0-P1-P2-P3
1	Y		Y	Y	P0	2	0	0	0	P0-P1-P2-P3
2	Y	Y	Y	Y	P0	3	0	0	0	P0-P1-P2-P3
3	Y	Y	Y	Y	P0	4	0	0	0	P1-P2-P3-P0
4	Y	Y	Y	Y	P1	0	1	0	0	P1-P2-P3-P0
5	Y	Y	Y	Y	P1	0	2	0	0	P1-P2-P3-P0
6	Y	Y	Y	Y	P1	0	3	0	0	P2-P3-P0-P1
7	Y		Y	Y	P2	0	0	1	0	P2-P3-P0-P1
8	Y		Y	Y	P2	0	0	2	0	P3-P0-P1-P2
9	Y			Y	P3	0	0	0	1	P0-P1-P2-P3
10	Y		Y	Y	P0	1	0	0	0	P0-P1-P2-P3
11			Y	Y	P2	1	0	1	0	P0-P1-P2-P3
12			Y	Y	P2	1	0	2	0	P0-P1-P3-P2

If the same system also contains two ports that only request at priority level 1, then the system behavior will be slightly altered. The addition of these 2 ports creates the second priority group structure that adds to the arbitration complexity. Table 34-208 describes this system. The blue text highlights the priority level change for P4 and P5. Again, for simplification, the command queue is considered to never be full and it is assumed that commands from ports 0, 1, 2 and 3 are only received at priority level 0. The behavior is shown in Table 34-209. Note that if any of the priority 0 ports (P0, P1, P2, P3) are requesting, the system behavior will match the behavior when there is only one priority group, as in Table 34-207. Ports 4 and 5 can only win arbitration when no higher-priority commands exist.

Table 34-208. System E Specification

Memory Controller bits	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5
AXIY_PRI0_RPRI	4	3	2	1	1	1
AXIY_PRI1_RPRI	1	1	1	1	3	2
AXIY_P_ODR	0	1	2	3	4	5

Table 34-209. System E Operation

Cycle	Ports Requesting						Arbitration Winner	Next Counter						Next Scan Order
	P0	P1	P2	P3	P4	P5		P0	P1	P2	P3	P4	P5	
														Priority 0: P0-P1-P2-P3 Priority 1: P4-P5
0			Y			Y	P2	0	0	1	0	0	0	P0-P1-P2-P3 P4-P5
1	Y		Y			Y	P0	1	0	1	0	0	0	P0-P1-P2-P3 P4-P5
2			Y			Y	P2	1	0	2	0	0	0	P0-P1-P3-P2 P4-P5
3	Y		Y		Y	Y	P0	2	0	0	0	0	0	P0-P1-P3-P2 P4-P5
4			Y		Y	Y	P2	2	0	1	0	0	0	P0-P1-P3-P2 P4-P5
5					Y	Y	P4	2	0	1	0	1	0	P0-P1-P3-P2 P4-P5
6		Y			Y	Y	P1	2	1	1	0	1	0	P0-P1-P3-P2 P4-P5
7					Y	Y	P4	2	1	1	0	2	0	P0-P1-P3-P2 P4-P5
8					Y	Y	P4	2	1	1	0	3	0	P0-P1-P3-P2 P5-P4
9					Y	Y	P5	2	1	1	0	0	1	P0-P1-P3-P2 P5-P4
10					Y		P4	2	0	1	0	1	1	P0-P1-P3-P2 P5-P4

34.6.3.7 Priority Relaxing

From Table 34-209, it is evident that ports at lower priority levels will not win arbitration in weighted round-robin arbitration unless there are no higher priority requests. This could mean that, in a situation where high priority requests are being received continuously, lower priority requests could be locked out indefinitely. To avoid this scenario and control the arbitration latency for lower-priority commands, it is possible to

disable priority groups temporarily. This is known as priority relaxing, and it is a time-controlled function. Each higher priority group will be temporarily disabled when the pre-set counter value for the lower priority group has been reached and a request is waiting. The *axiY_prirlx* bits set the counter value for port Y at which the priority relax condition will be triggered. The timing counters inside each port are controlled by the DDR_CR119[WRR_LATCTL]. When the latency control bit is set to 'b1, the timing counters are free-running. Any timing counter may hit its *axiY_prirlx* bits value at any point. When this occurs, higher-priority groups are disabled to allow a waiting request for this port to be processed. This results in a random latency for each port, but the maximum latency is fixed at the *axiY_prirlx* bits value. If the current port does not have any commands waiting when the timing counter hits the relax value, then the counter will be reset and the Arbiter will function normally. When the WRR_LATCTL bit is cleared to 'b0, the timing counters only count while that port has a waiting request that is not being processed. In this case, when the port's *axiY_prirlx* bits value is reached, all priority groups at priority levels higher than the waiting request are disabled. This port's command is granted arbitration and is moved through to the MC core. Since the priority relax parameters and counters are associated with individual ports, it is possible that multiple priority relax counters could reach their specified value simultaneously. In this case, the lower priority command will be arbitrated first and then the higher priority command. This situation could alter the arbitration latency slightly, causing it to be longer than the expected value in the priority relax parameter. Consider the System F as described in Table 34-210. The same conditions apply as for the previous example. The command queue is considered to never be full, commands from ports 0, 1, 2 and 3 are only received at priority level 0, and commands from ports 4 and 5 are only received at priority 1.

Table 34-210. System F Specification

Memory Controller bits	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5
AXIY_PRI0_RPRI	4	3	2	1	1	1
AXIY_PRI1_RPRI	1	1	1	1	2	1
AXIY_P_ODR	0	1	2	3	4	5

Table 34-211 shows the system behavior. The exact settings of the latency control and priority relax parameters are not shown. Instead, the "Relaxed Ports" column indicates which, if any, ports have hit their priority relax values. The following cycles are important to observe:

- Cycles 1 and 7: A port relaxes while a higher priority request and a higher scan order request are both present. The relaxed port still wins arbitration.

- Cycle 4: Two ports of the same priority relax. The higher scan order request wins arbitration.
- Cycle 5: Two ports of different priorities relax.

The lower priority port that relaxed wins arbitration. The higher priority port that relaxed will maintain its relax condition, and win arbitration in the next cycle.

Table 34-211. System F Operation with Priority Relaxing

Cycle	Ports Requesting						Relaxed Ports	Arbitration Winner	Next Counter						Next Scan Order
	P0	P1	P2	P3	P4	P5			P0	P1	P2	P3	P4	P5	
															Priority 0: P0-P1-P2-P3 Priority 1: P5-P4
0			Y		Y	Y		P2	0	0	1	0	0	0	P0-P1-P2-P3 P5-P4
1			Y		Y	Y	P5	P5	0	0	1	0	0	1	P0-P1-P2-P3 P4-P5
2			Y		Y	Y		P2	0	0	2	0	0	0	P0-P1-P3-P2 P4-P5
3		Y			Y	Y		P1	0	1	0	0	0	0	P0-P1-P3-P2 P4-P5
4	Y				Y	Y	P4,P5	P4	0	1	0	0	1	0	P0-P1-P3-P2 P4-P5
5	Y				Y	Y	P0,P5	P5	0	1	0	0	1	1	P0-P1-P3-P2 P4-P5
6	Y				Y		P0	P0	1	1	0	0	1	0	P0-P1-P3-P2 P4-P5
7	Y				Y	Y	P4	P4	1	1	0	0	2	0	P0-P1-P3-P2 P5-P4
8	Y	Y	Y			Y		P0	2	1	0	0	0	0	P0-P1-P3-P2 P5-P4

Table continues on the next page...

Table 34-211. System F Operation with Priority Relaxing (continued)

Cycle	Ports Requesting						Relaxed Ports	Arbitration Winner	Next Counter						Next Scan Order
	P0	P1	P2	P3	P4	P5			P0	P1	P2	P3	P4	P5	
9		Y	Y	Y		Y		P1	2	2	0	0	0	0	P0-P1-P3-P2 P5-P4
10		Y	Y	Y		Y	P2	P2	2	2	1	0	0	0	P0-P1-P3-P2 P5-P4

Priority relaxing allows low priority commands to be able to move through the Arbiter to the MC core. This will ensure that the system can meet maximum latency requirements.

34.6.3.8 Port Pairing

The Memory Controller Arbiter incorporates a feature which allows adjacent ports to be grouped together and considered jointly for arbitration. The DDR_CR120[W_RR_WSHR] controls this function, with 1 bit per pair of ports in the Memory Controller. Bit 0 controls ports 0 and 1, Bit 1 controls ports 2 and 3, etc. If the Memory Controller interfaces to an odd number of ports, the highest numbered port is excluded from the port pairing system. Since the ports are grouped together, their relative priorities are not considered separately. Referring to Understanding Relative Priority, the general formula for port priority allocation is the ratio of that port's relative priority bits (AXIY_PRIYZ_RELPRI) to the sum of all requesting port's relative priority values. In this case, the relative priority value of only one of the paired ports is used for the sum calculation. This means that the bandwidth will be divided differently among the ports. If the port pair is at the top of the scan order, and either of the ports is requesting, then the requesting port will win arbitration. If both are requesting, port ordering is used to determine which port wins arbitration. Note that when the ports are paired, their scan order can never be altered and they will always remain together in the scan order. Their counters increment together, and so when they reach their relative priority value, the port pair will dynamically be placed at the bottom of the scan order for that priority group.

In order for port weight sharing to be used, the relative priority parameters for the port pair must be programmed to the same value and the port order of the paired ports should be sequential. If either condition is not followed, an error bit will be set to 'b1.

Consider System G as described in Table 34-212. Again, for simplification, the command queue is considered to never be full, commands from ports 0, 1, 2 and 3 are only received at priority level 0 and commands from ports 4 and 5 are always at priority 1. However, now ports 0 and 1 and ports 4 and 5 are paired.

Table 34-212. System G Specifications

Memory Controller bits	Port 0	Port 1	Port 2	Port 3	Port 4	Port 5
AXIY_PRI0_RELPRI	4	3	2	1	1	1
AXIY_PRI1_RELPRI	1	1	1	1	2	1
AXIY_P_ODR	0	1	2	3	4	5
W_RR_WSHR	1 (Paired)		0 (Not Paired)		1 (Paired)	

Table 34-213 shows the system behavior with port pairing. Since ports 4 and 5 are still at a lower priority, they will be ignored unless none of the higher priority ports (P0, P1, P2 or P3) are requesting. Note the following points:

- When either port of a port pair wins arbitration, the counters for both ports of the pair increment.
- In Cycle 3, the port pair P0/P1 reaches its allocated relative priority. Note that the port pair dynamically moves to the bottom of the scan order.
- In Cycle 8, the port pair P4/P5 reaches its allocated relative priority. However, since these are the only requests at priority 1, the scan order does not change.

Table 34-213. System G Operation

Cycle	Ports Requesting P0	P1	P2	P3	P4	P5	Arbitration Winner	Next Counter P0	P1	P2	P3	P4	P5	Next Scan Order
														Priority 0: P0-P1-P2-P3 Priority 1: P5-P4
0	Y		Y				P0	1	1	0	0	0	0	P0-P1-P2-P3 P5-P4
1	Y		Y			Y	P0	2	2	0	0	0	0	P0-P1-P2-P3 P5-P4
2			Y			Y	P2	2	2	1	0	0	0	P0-P1-P2-P3 P5-P4

Table continues on the next page...

Table 34-213. System G Operation (continued)

Cycle	Ports Requesting						Arbitration Winner	Next Counter						Next Scan Order
	P0	P1	P2	P3	P4	P5		P0	P1	P2	P3	P4	P5	
3	Y	Y		Y		Y	P0	3	3	1	0	0	0	P2-P3-P0-P1 P5-P4
4		Y		Y		Y	P3	0	0	1	1	0	0	P2-P3-P0-P1 P5-P4
5		Y		Y		Y	P3	0	0	1	2	0	0	P2-P0-P1-P3 P5-P4
6		Y				Y	P1	1	1	1	0	0	0	P2-P0-P1-P3 P5-P4
7						Y	P5	1	1	1	0	1	1	P2-P0-P1-P3 P5-P4
8					Y		P4	1	1	1	0	2	2	P2-P0-P1-P3 P5-P4
9					Y	Y	P5	1	1	1	0	1	1	P2-P0-P1-P3 P5-P4
10		Y	Y		Y	Y	P2	1	1	2	0	1	1	P0-P1-P3-P2 P5-P4
11		Y			Y	Y	P1	2	2	0	0	1	1	P0-P1-P3-P2 P5-P4

34.6.3.9 Error Conditions

With the programming complexities of the weighted round-robin arbitration scheme, an error reporting mechanism is included to notify users of illegal programming scenarios. These error conditions will each set a bit in the DDR_CR120[WRR_ERR] to 'b1. The potential error conditions are:

- Bit 0 = The 2 *axiY_p_odr* bits do not all contain unique values.
- Bit 1 = Any of the *axiY_priz_relpri* bits have been programmed with a zero value. A 0 value leads to unknown behavior. The minimum allowable value is 1.

- Bit 2 = Any ports, whose related bit of the DDR_CR120[W_RR_WSHR] is set to 'b1, do not have the same values in their *axiY_priz_relpri* bits.
- Bit 3 = For ports whose related bit of the W_RR_WSHR bit is set to 'b1, the values of the *axiY_p_odr* bits are not sequential.

If bit 0, 2 or 3 is set to 'b1 in the WRR_ERR, and any of the ports are paired in the W_RR_WSHR, then all weight sharing data will be ignored during Memory Controller initialization and the ports will be prioritized by port number. If port pairing is not being used, but the bit 0 error condition is set to 'b1, then ports with a non-unique port ordering are prioritized by port number.

Note

The user is strongly cautioned against modifying the values of the port ordering or relative priority parameters during active port usage.

34.6.3.10 Programmable Options for Weighted Round Robin Arbitration

The Memory Controller's weighted round-robin arbitration scheme provides a great deal of programmable control for the user. The parameters are referenced throughout this chapter and are summarized here for clarity:

- *axiY_r_pri* (DDR117[AXI0_R_PRI] and DDR118[AXI1_R_PRI])
- *axiY_w_pri* (DDR117[AXI0_W_PRI] and DDR118[AXI1_W_PRI])
- *axiY_priz_relpri* ((DDR120[AXI0_PRI1_RPRI], DDR120[AXI0_PRI0_RPRI], DDR121[AXI0_PRI3_RPRI],DDR121[AXI0_PRI2_RPRI], DDR122[AXI1_PRI1_RPRI], DDR122[AXI1_PRI0_RPRI], DDR123[AXI1_PRI3_RPRI] and DDR123[AXI1_PRI2_RPRI])
- *axiY_p_odr* (DDR121[AXI0_P_ODR] and DDR123[AXI1_P_ODR])
- *axiY_prirlx* (DDR122[AXI0_PRIRLX] and DDR124[AXI1_PRIRLX])
- DDR119[WRR_LATCTL]
- DDR120[W_RR_WSHR]
- DDR120[WRR_ERR]

34.6.4 Core Command Queue with Placement Logic

The MC core contains a command queue that accepts commands from the Arbiter. This command queue uses a placement algorithm to determine the order that commands will be placed into the command queue. The placement logic follows many rules to determine where new commands should be inserted into the queue, relative to the contents of the command queue at the time. Placement is determined by considering address collisions, source collisions, data collisions, command types and priorities. In addition, the placement logic attempts to maximize efficiency of the MC core through command grouping, write-to-read splitting and bank splitting. Many of the rules used in placement may be individually enabled/disabled. In addition, the command queue may be disabled by clearing the DDR_CR75[PLEN] bit, resulting in an in-line queue that services requests in the order they are received. If the DDR_CR75[PLEN] is cleared to 'b0 and the DDR_CR79[INODR_ACT] is set to 'b1, the placement algorithm will be ignored.

34.6.4.1 Rules of the Placement Algorithm

The factors affecting command placement all work together to identify where a new command fits into the execution order. They are listed in order of importance.

34.6.4.1.1 Address Collision/Data Coherency Violation

The order in which read and write commands are processed in the memory controller is critical to proper system behavior. While reads and writes to different addresses are independent and may be re-ordered without affecting system performance, reads and writes that access the same address are significantly related. If the port requests a read after a write to the same address, then repositioning the read before the write would return the original data, not the changed data. Similarly, if the read was requested ahead of the write but accidentally positioned after the write, then the read would return the new data, not the original data prior to being overwritten. These are significant data coherency mistakes.

To avoid address collisions, reads or writes that access the same chip select, bank and row as a command already in the command queue will be inserted into the command queue after the original command, even if the new command is of a higher priority. This rule is ignored when comparing a new read command to an existing read. Even if an address collision occurs between these reads, there is no data integrity issue and the data may be returned in any order.

This factor may be enabled/disabled through the DDR_CR74[ADDR_CMP_EN] and should only be disabled if the system can guarantee coherency of reads and writes.

34.6.4.2 Source ID Collision

Each port is assigned a specific source ID that is a combination of the port and thread ID information, and identifies the source uniquely. This allows the memory controller to map data from/ to the correct source/destination.

Note that a source ID does contain port identification information which means that the rules for placement are dependent on the requesting port. There will not be source ID collisions between ports.

In general, read commands from the same source ID will be placed in the command queue in order. Therefore, a read command with the same source ID as a read command already in the command queue will be processed after the original read command. All write commands from a port, even with different source IDs, will be executed in order.

The behavior of commands of different types from the same source ID is dependent on the user configuration. For this Memory Controller, the placement of new read/write commands that collide in terms of source ID with existing entries in the command queue will only depend on other commands of the same type, not on different types. This means that, if there are no address conflicts, a read command could be executed ahead of a write command with the same source ID, and likewise a write command could be executed ahead of a read command with the same source ID.

This feature will always be enabled.

34.6.4.3 Priority

Priorities are used to distinguish important commands from less important commands. Each command is given a priority based on the command type through the programmable bits DDR_CR117[AXI0_R_PRI] and DDR_CR118[AXI1_W_PRI].

The placement algorithm will attempt to place higher priority commands ahead of lower priority commands, as long as they have no source ID or address collisions. Higher priority commands will be placed lower in the command queue if they access the same address, are from the same requestor or use the same buffer as lower priority commands already in the command queue.

This feature is enabled through the DDR_CR75[PRI_EN].

34.6.4.4 Bank Splitting

Before accesses can be made to two different rows within the same bank, the first active row must be closed (pre-charged) and the new row must be opened (activated). Both activities require some timing overhead; therefore, for optimization, the placement logic will attempt to insert the new command into the command queue such that commands to other banks may execute during this timing overhead. The placement of the new commands will still follow priority, source ID and address collision rules.

This feature is enabled through the DDR_CR74[BANKSPLT_EN].

34.6.4.5 Write-to-Read Splitting

When a read command follows a write command to the same chip select, there is some timing overhead to switch command types. For optimization, the placement logic will attempt to insert the new command into the command queue to separate two commands addressing the same chip select of different types where the write is going to execute before the read. The placement of the new commands will still follow priority, source ID and address collision rules. This feature is enabled through the DDR_CR76[W2R_SPLT_EN].

34.6.4.6 Read/Write Grouping

The memory suffers a small timing overhead when switching from read to write mode. For efficiency, the placement logic will attempt to place a new read command sequentially with other read commands in the command queue, or a new write command sequentially with other write commands in the command queue. Grouping will only be possible if no priority, source ID or address collision rules are violated. This feature is enabled through the DDR_CR75[RW_EN].

34.6.4.6.1 Bank Conflicts and Read/Write Grouping

If the new command addresses the same chip select and same bank, but a different row, as a command currently in the command queue, these commands are considered to have a bank conflict. As described in Bank Splitting, the placement logic will attempt to separate commands with bank conflicts. For this memory controller, certain placements are prohibited for read/write grouping to support ideal bank splitting.

These checks are controlled through the DDR_CR76[D_RW_G_BKCN]. If bit [0] of this parameter is set to 'b1, a new command will be prohibited from placement in the entry directly before or directly after the command with a bank conflict. If bit [1] of this parameter is also set to 'b1, the new command will also be prohibited from being placed two entries before or two entries after the command with a bank conflict. The following table shows a simplified command queue.

Table 34-214. Simple Command Queue Example

PQ Entry	Read/write	Bank	Row
0	Rd	0	0
1	Rd	0	0
2	Rd	0	0
3	Rd	1	0
4	Rd	0	0
5	Rd	0	0

For this example, a new entry is received that is a READ to BANK 1, Row 1. Assume that no priority, source ID or address collision rules are violated. This new command would have a bank conflict with PQ Entry 3.

- If D_RW_G_BKCN [0] = 'b1, the new command could not placed immediately before or after the conflicting command. If the command was placed into entry 3, entries 3-5 would be moved to entries 4-6, and the bank conflicts would occur between entries 3 and 4. If the command was placed into entry 4, entries 4-5 would be moved to entries 5-6 and the bank conflict would still occur between entries 3 and 4. Therefore, entries 3 and 4 are prohibited for placement.
- If D_RW_G_BKCN [1] = 'b1, the new command could not placed two entries before or two entries after the conflicting command. If the command was placed into entry 2, entries 2-5 would move to entries 3-6, and the bank conflict would occur between entries 2 and 4. If the command was placed into entry 5, the bank conflict would occur with between entries 3 and 5. Therefore, entries 2 and 5 are also prohibited for placement.
- Therefore, if the D_RW_G_BKCN bit was set to 'b11, the new entry could only be placed at entry 0, 1 or 6, allowing at least 2 commands in between the conflicting commands.

Note

It is not meaningful to set bit [1] of the DDR_CR76[D_RW_G_BKCN] without bit [0].

34.6.4.6.2 Chip Select Grouping with Read/Write Grouping

When attempting to group read and write commands, the placement logic will also consider the chip select for the commands. If possible, read commands will be grouped with read commands to the same chip select, and write commands with write commands to the same chip select. If chip select grouping is not possible, commands will still be grouped by command type if the DDR_CR75[RW_EN] is set to 'b1. If read/write grouping is disabled (DDR_CR75[RW_EN] is cleared to 'b0), chip select grouping will have no effect. This feature is enabled through the DDR_CR76[CS_EN].

34.6.4.6.3 Page Grouping with Read/Write Grouping

When attempting to group read and write commands, the placement logic will also consider the page for the commands. If possible, read commands will be grouped with read commands to the same page, and write commands with write commands to the same page. If page grouping is not possible, commands will still be grouped by command type if the DDR_CR75[RW_EN] is set to 'b1. If read/ write grouping is disabled (DDR_CR75[RW_EN] is cleared to 'b0), page grouping will have no effect. This feature is enabled through the DDR_CR75[RW_PG_EN] parameter.

34.6.4.7 Command Execution Order After Placement

Once a command has been placed in the command queue, selection logic will be used to determine how to pull commands from the queue for execution. This logic may be disabled by setting the DDR_CR79[INODR_ACT] to 'b1, resulting in the command queue executing the commands in the order that they are placed relatively in the command queue. If the DDR_CR79[INODR_ACT] is cleared to 'b0, the selection logic will be utilized. Regardless of the setting of this parameter, high-priority command swapping and command aging are provided which may affect commands after they have been placed into the command queue.

34.6.4.7.1 Command Selection Logic

On each clock cycle, the selection logic will scan the top 4 entries of the command queue to determine which command to execute. This value is defined at configuration. Commands are considered for execution based on bank readiness, availability of at least 1 burst of data (writes), availability of storage for at least 1 burst of data (reads), bus turnaround timing (JEDEC-specified and programmable) and conflicts. Similar to the placement rules, a command will not be executed before a command that was placed ahead of it in the command queue if there are any address, source ID or bank conflicts.

All placement rules mentioned in this chapter are followed by the selection logic other than priority. It is possible that lower priority commands may be executed ahead of higher priority commands if the higher priority commands are not ready to execute, provided that there are no conflicts with commands ahead in the command queue. The memory controller will also not execute a read/ modify/write sequence before another read/modify/write sequence ahead of it in the command queue due to limited storage in the core.

The selection feature is disabled through the DDR_CR79[INODR_ACT] . If this bit is set to 'b1, only the top entry of the command queue will be considered for execution. The following figure shows the command selection logic relative to the rest of the placement logic.

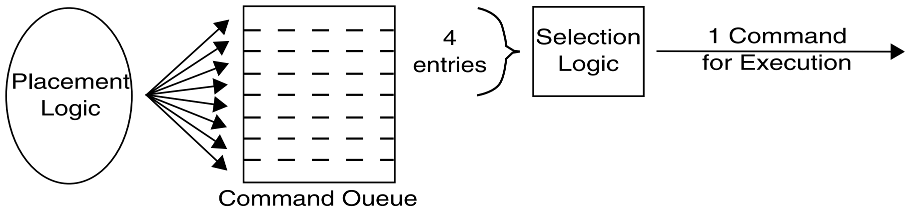


Figure 34-195. Selection logic

34.6.4.7.2 High-Priority Command Swapping

Commands are assigned priority values to ensure that critical commands are executed more quickly in the memory controller than less important commands. Therefore, it is desirable that high-priority commands pass into the MC core as soon as possible. The placement algorithm takes priority into account when determining the order of commands, but still allows a scenario in which a high-priority command sits waiting in the command queue while another command, perhaps of a lower priority, is in process. The high-priority command swapping feature allows this new high-priority command to be executed more quickly. If the user has enabled the swapping function through the DDR_CR79[SWAP_EN] parameter, then the behavior of the swapping logic will be dependent on the value of the DDR_CR79[INODR_ACT]. If the command queue must be executed in order (INODR_ACT = 'b1), the entry at the top of the command queue will be compared with the current command in progress. If the selection logic is being used (INODR_ACT = 'b0), the top 4 entries of the command queue will be compared with the current command in progress to determine which command may need to be executed first. If the selected command queue entry is of a higher priority (not the same priority), from a different ID, and it does not have an address or source ID conflict with the current command being executed, then the original command will be interrupted. If the command is to be interrupted, it will be halted after completing the current burst, stored and placed at the top of the command queue, and the new command will begin executing. As long as the command queue is not full, new commands may continue to be

inserted into the command queue based on the placement rules, even at the top of the queue ahead of the interrupted command. The selection logic will determine the command to execute next. Whenever the interrupted command is resumed, it will start from the point at which it was interrupted. Note that priority 0 commands will never be interrupted, so the user should set any commands that should not be interrupted to priority 0.

34.6.4.7.3 Command Aging

Since commands can be inserted ahead of existing commands in the command queue, the situation could occur where a low priority command remains at the bottom of the queue indefinitely. To avoid such a lockout condition, aging counters have been included in the placement logic that measure the number of cycles that each command has been waiting. If an aging counter hits its maximum, the priority of the associated command will be decremented by one (lower priority commands are executed first). This increases the likelihood that this command will be picked by the selection logic and execute. Note that this command does not move relative positions in the command queue when it ages; the new priority will be considered when placing new commands into the command queue.

Aging is controlled through a master aging counter and command aging counters associated with each command in the command queue. The DDR_CR74[AGE_CNT] and DDR_CR74[CMD_AGE_CNT] hold the initial values for each of these counters, respectively. When the master counter counts down the AGE_CNT bit value, a signal is sent to the command aging counters to decrement. When the command aging counters have expired, the priority of the associated command is decremented and the counter is reset. Therefore, a command does not age by a priority level until the total elapsed cycles has reached the product of AGE_CNT bit value+ 1 and CMD_AGE_CNT bit value + 1. The maximum number of cycles that any command can wait in the command queue until reaching the top priority level is the product of the AGE_CNT bit value +1, the CMD_AGE_CNT bit value +1, and the number of priority levels in the system.

If command swapping is enabled, it is possible that a command in progress could be interrupted by a higher priority command in the command queue. This situation could arise if a new higher-priority command arrives and is placed while the current command is in progress, or by a command in the queue aging to a lower priority while the current command is in progress. An interrupted command will be placed at the top of the command queue.

This feature will always be enabled.

34.6.4.8 ACT Request Control

The Memory Controller provides a means to limit which commands of the command queue may issue ACT requests. This can be used to prevent situations in which an ACT is issued for a command in the queue, and before that command can be executed, a new command is placed ahead of it which accesses the same bank but a different row. This would require a PRE-ACT sequence that may have been avoided if the first ACT was never issued.

This functionality is controlled through the DDR_CR76[NQENT_ACTDIS], which specifies the number of entries of the command queue in which ACT requests are not allowed. For this 8-deep command queue, the entries are numbered 0-7, where entry 0 is the command next to execute.

Table 34-215. Programming of the NQENT_ACTDIS bits

Value	Effect
0x0	ACT request can occur from entries 0-7
0x1	ACT request can occur from entries 0-6
0x2	ACT request can occur from entries 0-5
0x3	ACT request can occur from entries 0-4
0x4	ACT request can occur from entries 0-3
0x5	ACT request can occur from entries 0-2
0x6	ACT request can occur from entries 0-1
0x7	ACT request can occur from entries 0

34.6.5 ECC Options

Memory Controllers provide an optional error reporting and correcting circuitry that can be used to verify data in memory and correct memory errors if they occur. The logic will check for errors in both the data and the check code on all read transactions.

ECC, or error checking and correcting, is the process of detecting bit errors in the memory data and if possible, correcting them. This function can confirm the accuracy of data and remove or at least identify bit errors.

ECC works by storing unique "check codes" in memory. A check code is a mathematical description of the information in an aligned segment of memory known as an "ECC data word". The check code is always related to the entire ECC data word, and is used inside the Memory Controller on all memory reads to control data accuracy. Check codes are not input from, or output to, the user interface.

An ECC data word can not start and end at any random address; these words are memory aligned to their size. The starting addresses of ECC data words are defined as ECC word boundaries and the alignment of user transactions to these boundaries determines how transactions are processed inside the Memory Controller. If the user does not wish to use the ECC option, the ECC module may be disabled by setting the control bit DDR_CR57[CTRL_RAW] to 'b10.

This Memory Controller supports a 32-bit ECC data word size. A 7-bit check code is maintained for each 32-bit memory area. ECC word boundaries fall on each 4-byte address (0xN0, 0xN4, 0xN8, 0xNc).

34.6.5.1 Initialization of memory when using ECC

When using ECC, or error checking and correcting,additional steps are required to configure the memory and all ECC values after the controller has been initialized.

The entire memory region that will be used should be written once. This write is required so that a correct ECC tag can be written for each memory location. There are two recommended methods for performing this initialization.

- Using the DMA to initialize the memory.

  The DMA controller is usually the most efficient means of writing the initial data and ECC values to the memory. The DMA should be configured to perform a 32-byte write to DDR destination addresses with the NBYTES field configured so that any memory locations that will be used by the system are initialized.

- Using the CPU to initialize the memory.

  A CPU can also be used to write the initial data and ECC values to memory; however, when using a CPU to perform the writes the accesses can actually trigger ECC errors. To avoid this, set the DDRMC_CR058[ECC_DIS_WCRER] bit, then perform the writes to set the initial values. After all of the memory locations have been initialized, the ECC_DIS_WCRER bit can be cleared.

34.6.6 Low Power Operation

The Memory Controller provides various user-configurable low power options to address power savings.

There are seven low power states available in the Memory Controller and managed by the LPC module. The low power states are listed from least to most power saving.

Note

Transitions will only be made into deeper low power states. If the user requires to switch to a higher power state, the current state must be exited and then the new state entered.

Note

Low power state transitions that may be completed without a low power exit will be performed when possible. The system will include low power exits if necessary when switching from one low power state to another. Low power modes entry exit is controlled through the register interface

1. Active Power-Down

   The memory controller sets the memories into power-down while any row is active in the bank. This state reduces the overall power consumption of the system, but has the least effect of all the low power states. In this state, the memory controller and memory clocks are fully operational, but the CKE input bit to the memories is de-asserted. If entry into the "Active Power-Down" state was requested, the memory will enter either active or pre-charge power-down mode depending on the state of the rows. If there are no open rows, pre-charge power-down mode will be entered. If the DDR_CR34[LP_REFEN] is set to 'b1, the memory controller will continue to monitor memory refresh needs and will automatically bring the memory out of power-down to perform these refreshes. When a refresh is required, the CKE input bit to the memories will be re-enabled. This action brings the memories out of power-down. Memory that has been transitioned into active power-down mode will automatically transition to pre-charge power-down mode after the memory controller performs a refresh since the refresh process includes a pre-charge all command.

2. Active Power-Down with Memory Clock Gating

   The memory controller sets the memories into power-down while any row is active in the bank, and gates off the clock to the memories. Refreshes will be handled as in the "Active Power-Down" state, with the exception that gating on the memory clock will be removed before asserting the CKE pin. Memory that has been transitioned into active power-down mode will automatically transition to pre-charge power-down mode (with the clock gated) when the memory controller performs a refresh since the refresh process includes a pre-charge all command. Before the memories are removed from power-down, the clock will be gated on again.

This low power state is only valid for LPDDR2 memory. The user should not use this state for memories that do not support memory clock gating. Clock gating is not supported for standard DDR3 memories. When set into this state, the memory controller will attempt to place the memories in power-down and gate off the memory clock. The memory will function unpredictably and may hang.

3. Pre-Charge Power-Down

   The memory controller sets the memories into power-down once all banks are idle. If any rows are active, prior to issuing the power-down mode command, the memory controller will issue a pre-charge all command. If the DDR_CR34[LP_REFEN] is set to 'b1, the memory controller will continue to monitor memory refresh needs and will automatically bring the memory out of power-down to perform these refreshes. When a refresh is required, the CKE input bit to the memories will be re-enabled. This action brings the memories out of power-down. Once the refresh has been completed, the memories will be returned to pre-charge power-down mode following the de-assertion the CKE input bit. For DDR3, pre-charge power-down mode supports both fast and slow exit modes depending on how the memory mode registers are programmed. If the pre-charge power-down bit (MR0 [A12]) is cleared to 'b0 (slow exit), the timing parameter *txpdll_fN* will define the exit time from this low power state to a read command. If the pre-charge power-down parameter is set to 'b1 (fast exit), the timing parameter *tpdex_fN* will be used.

4. Pre-Charge Power-Down with Memory Clock Gating

   The memory controller sets the memories into power-down once all banks are idle, and gates off the clock to the memories. Refreshes will be handled as in the "Pre-Charge Power-Down" state, with the exception that gating on the memory clock will be removed before asserting the CKE pin. After the refresh has been completed, the memories will be returned to pre-charge power-down mode with the clock gated. Before the memories are removed from power-down, the clock will be gated on again. This low power state is only valid for LPDDR2 memory.

   The user should not use this state for memories that do not support memory clock gating. Clock gating is not supported for standard DDR1, DDR2 or DDR3 memories. When set into this state, the memory controller will attempt to place the memories in power-down and gate off the memory clock. The memory will function unpredictably and may hang.

5. Self-Refresh

The memory controller sets the memories into self-refresh mode. In this low power mode, the memory controller and memory clocks are fully operational and the CKE input bit to the memories is de-asserted. Since the memory automatically refreshes its contents, the memory controller does not need to send explicit refreshes to the memory.

6. Self-Refresh with Memory Clock Gating

   The memory controller sets the memories into self-refresh and gates off the clock to the memories. Before the memories are removed from self-refresh, the clock will be gated on again. This low power state is only valid for LPDDR2 memory.

7. Self-Refresh with Memory and Controller Clock Gating

   This is the deepest low power state of the memory controller. The memory controller sets the memories into self-refresh and gates off the clock to the memories. In addition, the clock to the memory controller will be gated off.

   Before the memories are removed from self-refresh, the memory controller and memory clocks will be gated on. If automatic exit from this state is enabled, a new transaction that addresses a memory device in this state will wake up the memory to process the transaction. This low power state is only valid for LPDDR2 memory.

Note

This state should not be entered when a read or write command is being processed. The user should ensure that the controller is idle before requesting entry into this state by checking the DDR_CR79[CTLBUSY].

Note

When the controller clock is gated through the low power control module, writes to any of the programmable command parameters - parameters that result in command execution within the controller - will be ignored and not result in execution of the associated commands. Any commands issued while the controller clock is gated may or may not be executed once the controller clock is un-gated. In general, command registers should not be programmed during controller clock gating as the commands will have undefined results.

34.6.6.1 Automatic Interface

The LPC module supports automatic entry into each of the low power states based on programmable enables and idle state monitors. Each low power state has a separate enable bit and counter. As with the external pin interface and software programmable interface, the automatic interface must win arbitration of the LPC module to issue requests. The automatic interface has the lowest priority for arbitration.

When the memory controller is idle, each of the seven timing counters that are enabled begin counting down the cycles of inactivity. Idle time requires that no read or write commands are executing or pending in the memory controller core command queue or any of the ports. For the power-down states, idle time begins when no commands are waiting to be sent to memory. For the self-refresh states, idle time begins only after all the read data for outstanding read commands has been retrieved; this restriction ensures that all read data is received even if the automatic request includes gating off the memory controller clock.

34.6.6.1.1 Automatic Entry

If any of the counters expire, the automatic interface will request arbitration. The timing counters are initially loaded with the values in the associated parameters, and will only decrement if the associated DDR_CR36[LPAUTO] parameter bit is set to 'b1 and the counters are loaded with a non-zero value. If no other interface has control of the LPC module (DDR_CR35[LP_ARBST] = 'b00 or 'b11), the automatic interface will win arbitration and the specified low power state will be compared to the current state of the memories. If the memory is already in a low power state, and the expired counter is associated with a deeper low power state than the current state, the LPC module will trigger an entry into the new low power state. If the expired counter is associated with a higher power usage state, the counter expiration will be ignored.

If a counter expires while the LPC module is being controlled by the external pin interface or the software programmable interface, the automatic request will remain pending until the arbiter returns to idle (LP_ARBST = 'b00) and the automatic interface is granted control of the LPC module. A pending request will also be cancelled if a read or write command enters the Memory Controller core command queue, resetting the automatic counters to the programmed values.

Multiple automatic low power idle counters may expire at the same time. When this happens, the counter associated with the deepest low power state will be entered. No state change will occur if current low power state is deeper than the states associated with any of the expired counters.

34.6.6.1.2 Automatic Exit

The automatic interface also supports automatic exit from a low power state if the system requires, with separate enable bits for each low power state. During an automatic exit, all of the idle counters are reset to their programmed values and the memories are returned to normal operation. When a new read or write command enters the memory controller command queue, if the current low power state's associated DDR_CR36[LP_AEXEN] is set to 'b1, the automatic interface will request arbitration. If no other interface has control of the LPC module ((DDR_CR35[LP_ARBST] = 'b00 or 'b11), the automatic interface will win arbitration and a low power exit will be triggered. If the current state is not enabled for automatic exit in the DDR_CR36[LP_AEXEN], the LPC module will not exit low power.

If the LPC module is being controlled by the external pin interface or the software programmable interface when the new read or write command appears, the automatic request will remain pending until the arbiter returns to idle ((DDR_CR35[LP_ARBST] = 'b00) and the automatic interface is granted control of the LPC module.

Only new read or write commands will cause the counters to be reloaded to their programmed values and trigger an exit. Other commands, including MRR, MRW, low power entry and exit commands, register accesses, refresh, ZQ, etc., do not reset the counters and do not trigger an exit from low power. When the memory controller is idle and the memories have automatically entered a low power state, these commands may be prevented from executing.

It is generally expected that any state that is defined for automatic entry DDR_CR36[LPAUTO] should also be enabled for automatic exit DDR_CR36[LP_AEXEN]. Automatic exit may also be used for low power states that are not defined for automatic entry, but are expected to be entered manually through the software programmable interface DDR_CR35[LP_CMD]. If it is desirable to use both manual and automatic entry/ exit into the same low power state, then the user may need to re-program the DDR_CR36[LP_AEXEN] parameter prior to issuing a request through the software programmable interface.

34.6.6.2 Refresh Masking

Regular refresh commands will be issued at the same intervals while the memory controller is operating normally, is idle, or is in any of the low power states. However, for memory arrays with multiple chip selects, the Memory Controller supports the ability to mask refreshes while in any of the power-down low power states. By setting bits of the DDR_CR34[LP_REFEN] parameter to 'b1, auto-refreshes will be masked for the associated chip selects.

It is the user's responsibility to ensure that refreshes are not constantly masked, and that each chip select is refreshed periodically.

34.6.6.3 Mobile DDR SDRAM Memories

34.6.6.3.1 Enabling Mobile Usage

When using a mobile memory, bit 0 of the DDR_CR00[DRAM_CLASS] must be set to 1. This enables the memory controller to use the initialization sequence and mode register addressing appropriate to mobile memories. When bit 0 of the DDR_CR00[DRAM_CLASS] is cleared to 'b0, a standard DDRSDRAM or SDRAM memory may be used.

34.6.6.3.2 Partial Array Self-Refresh

For mobile memories, the Memory Controller is capable of supporting refreshes to subsections of the memory array. To facilitate this capability, separate parameters are provided to supply the mode register data for each chip select. These parameters are named *mrY_data_X*, where X represents the chip select and Y represents 0, 1, 2 or 3 - depending on the memory type being used.

Having separate control parameters for the mode register data allows the individual chips to set their own masked refresh. The DDR_CR45[WRMD] controls the writing of this mode register data into the registers. When the DDR_CR45[WRMD] is set to 'b1 initially, the mode register of chip select 0 will be written. Each subsequent setting of the DDR_CR45[WRMD] to 'b1 will write the mode register of the next chip select (1, 2, 3, etc.).

Note that the memory controller does not check if operations attempt to access addresses outside of the refresh ranges set by the memory mode registers. Any accesses to these addresses may result in corrupt or lost data.

34.6.7 Out-of-Range Address Checking

It is possible that the master attempts to write to an invalid address. For this reason, all incoming addresses are always checked against the addressable physical memory space. If a transaction is addressed to an out-of-range memory location, then bit 1 of the DDR_CR80[INT_STAT] will be set to 'b1 to alert the user of this condition. The memory controller will record the address, source ID, length and type of transaction that caused the out-of-range interrupt in the DDR_CR83[OORAD], DDR_CR84[OORID], DDR_CR84[OORLEN] and DDR_CR84[OORTYP].

Reading the out-of-range parameters will initiate the Memory Controller to empty these parameters and allow them to store out-of-range access information for future errors. The interrupt should be acknowledged by setting bit 1 of the DDR_CR81[INT_ACK] parameter to 'b1, which will in turn cause bit 1 of the DDR_CR80[INT_STAT] to be cleared to 'b0.

If a second out-of-range access occurs before the first out-of-range interrupt is acknowledged, then bit 2 of the DDR_CR80[INT_STAT] will be set to 'b1 to indicate that multiple out-of-range accesses have occurred. If the out-of-range parameters have been read when the second out-of-range error occurs, then the details for this transaction will be stored in the out-of-range parameters. If they have not been read, then the details of the second error will be lost.

Even though the address has been identified as erroneous, the Memory Controller will still process the read or write transaction. A read transaction will return random data which the user must receive to avoid stalling the memory controller.A standard, non-exclusive write transaction will write the associated data to an unknown location in the memory array, potentially over-writing other stored data. A command can not be aborted once accepted into the Memory Controller.

A special situation occurs with AXI exclusive accesses. For an exclusive write access only, if the exclusivity check fails, then the command will automatically be flushed internally. This is the only situation in which the erroneous data is NOT written to memory but is cleared out of the memory controller FIFOs. The DDR_CR84[OORTYP] bits [5:4] maintain status on AXI exclusive accesses.

Table 34-216. Out of Range Access Parameter

Memory Controller bit	Description
DDR_CR83[OORAD][31:0] Out of range address	Transaction Address
DDR_CR84[OORID] [13:0] Out of range source id	Bits [13:13] = Port ID Bits [12:0] = AXI Thread ID
DDR_CR84[OORLEN] [6:0] Out of range length	Total byte count of the transaction. • For write commands: (AXI port Y encoded write command length + 1) x $2^{(\text{AXI port Y encoded write command size})}$. • For read commands: (AXI port Y encoded read command length + 1) x $2^{(\text{AXI port Y encoded read command size})}$.

Table continues on the next page...

Table 34-216. Out of Range Access Parameter (continued)

Memory Controller bit	Description
DDR_CR84[OORTYP] [5:0] Out of range type	• 'b000000 = Non-Exclusive Write • 'b000001 = Non-Exclusive Read • 'b000010 = Non-Exclusive Masked Write • 'b000100 = Wrapped Write • 'b000101 = Wrapped Read • 'b000110 = Wrapped Masked Write • 'b001000 = Exclusive Write • 'b001001 = Exclusive Read • 'b001010 = Exclusive Masked Write • 'b010000 = Flushed Write • 'b100000 = Non-Exclusive Write with Auto-Precharge • 'b100001 = Non-Exclusive Read with Auto-Precharge • 'b100010 = Non-Exclusive Masked Write with Auto- Precharge • 'b100100 = Wrapped Write with Auto-Precharge • 'b100101 = Wrapped Read with Auto-Precharge • 'b100110 = Wrapped Masked Write with Auto-Precharge • 'b101000 = Exclusive Write with Auto-Precharge • 'b101001 = Exclusive Read with uto-Precharge • 'b101010 = Exclusive Masked Write with Auto-Precharge • 'b110000 = Flushed Write with Auto-Precharge • All other settings Reserved

34.6.8 Command to Command Timing

For flexibility and maximum user control, the Memory Controller provides a set of timing parameters to control the bus turnaround timing between commands of different types. In many systems, the extended pad or board delays may cause interference with the timings defined by default when changing chip selects or command types. These parameters allow the user to add additional cycles between a particular command combination by increasing the associated command to command parameter value based on the application-specific delay implementation. Each of the command to command parameters listed below can be changed individually to achieve timing which does not cause bus contention. The user should program each parameter with the actual number of clocks of turn-around required for this transaction combination on the data bus.

The memory controller accounts for the burst length and memory timing requirements. These command to command parameters do not compensate for these delays, but rather should be used to account for any additional delay needed such as variance in the pads, board or system. Also, more delay may be needed to turn the bi-directional DQ/DQS bus around for commands of different types for example if pad enable and disable times are different and additional time may be required to complete the turnaround. Or more time may be needed to account for variance in delays between different ranks. The "samecs" parameters may also be used to account for termination timing.

These command to command parameters should also be used to account for any constants in the timing equations in the JEDEC specification. For example, the LPDDR2 specification lists the R2W same CS delay as:

RL + tDQSCK_MAX + BL/2 + 1 - WL

The memory controller will account for all of these terms, other than the "+1". Therefore, for this memory system, the R2W_SMCSDL parameter would need to be programmed to at least a value of 0x1.

The command to command bits are:

- DDR_CR89[AODT_WRSMCS]
- DDR_CR89[AODT_RWSMCS]
- DDR_CR91[R2R_SMCSDL]
- DDR_CR91[R2W_SMCSDL]
- DDR_CR91[W2R_SMCSDL]
- DDR_CR92[W2W_SMCSDL]

34.6.9 Writing Mode Registers

The Memory Controller provides several options to program the mode registers in the memories. This functionality is controlled through the DDR_CR45[WRMD]. The encoding of this parameter defines which mode register(s) and which chip select(s) will be written. When the write has been completed (or if an invalid request was programmed), the mode register write complete interrupt (bit 21) will be set in the DDR_CR80[INT_STAT]. Any errors will be reported in the DDR_CR46[MRW_STAT] parameter. This parameter will be reset to 0x0 on the cycle after it is changed and can not be reprogrammed until the interrupt occurs.

34.6.9.1 WRMD (write mode register) bit fields

The fields of the DDR_CR45[WRMD] are:

- Bit [25] = Trigger the MRW sequence with the settings defined in bits [24:0].
- Bit [24] = Write all chip selects
  - 'b0 = Only the chip select identified in bits [15:8] will be written for the mode register(s) specified in bits [23:16].
  - 'b1 = Bits [15:8] will be ignored and all chip selects will be written for the mode register(s) specified in bits [23:16].
- Bits [23:16] = Mode register write type. Only one of these bits should be set when bit [25] is set. If no bits are set, an error will be flagged in the DDR_CR46[MRW_STAT]. If bit [24] is set to 'b1, all chip selects will be programmed. If bit [24] is cleared to 'b0, bits [15:8] will specify which chip select will be programmed.

  As shown in Table 34-217, the memory controller may provide mode register parameters that have no meaning for a particular memory system or chip select. If the user attempts a write to a mode register that is not relevant to the memory system for which the MC is programmed (through the DDR_CR00[DRAM_CLASS] parameter), the memory controller will only perform the write when applicable. For example, an all-chip select write to MR16 and MR17 will only write these mode registers when programmed for LPDDR2 memory systems.

  - Bit [23] = Write a single MRz

  The mode register specified in bits [7:0] will be written with the data in the associated mrsingle_data_X parameters. Even if an associated mrZ_data_X parameter exists (refer to Table 34-217), the data from that parameter will NOT be used for the write and the mrZ_data_X parameter will not be updated with this command. This setting can be used to write mode registers for which the Memory Controller does not provide a specific parameter.

  - Bit [18] = Write MR16, MR17
  - Bit [17] = Write MR0, MR1, MR2, MR3
  - All other bits reserved

- Bits [15:8] = Chip select number to be written. This field is only valid when bit [24] is cleared to'b0 specifying that only a single MRz will be written.
- Bits [7:0] = Mode register number to be written. This field is only valid when bit [23] is set to 'b1 specifying that only a certain MRz will be written.

  The mode register write(s) will be issued when the trigger bit (bit [25]) is set. Once the memory controller has completed all of the writes, the mode register write complete interrupt (bit 21) will be set in the DDR_CR80[INT_STAT] and errors will be reported in the DDR_CR46[MRW_STAT]. The DDR_CR45[WRMD] will be reset to 0x0 on the cycle after it is changed.

  - Bit [16] = Write MR0, MR1, MR2, MR3, MR16, MR17

34.6.9.2 MRW Module and Arbiter

The mode register write (MRW) module initiates the required mode register writes for any memory type. These MRWs are initiated from several modules including the initialization state machine, frequency scaling logic, low power logic and software. Since requests originate from different sources, multiple MRW requests may occur at the same time. To manage these requests, the MRW module contains an arbiter. If multiple requests occur, they will be serviced in the following priority order:

- ZQ Calibration
- Hardware Frequency Changes (MRW to MR0, MR1 and MR2 in order for each memory)
- Power-on initialization
- Power-on MR0 for initialization of non-LPDDR2 memories
- Power-on DLL reset

If a lower priority request is being serviced when a higher priority request is issued, the lower priority request will complete entirely before the higher priority request will be serviced. This is true even if the request must write to multiple mode registers and/or multiple chip selects. Software will initiate MRWs by programming the DDR_CR45[WRMD].

34.6.9.3 Programming Errors

If the DDR_CR45[WRMD]r was programmed incorrectly, the memory controller state machine may or may not issue the request. In either case, the mode register write complete interrupt (bit 21) will be set to 'b1 in the DDR_CR80[INT_STAT] and then the DDR_CR46[MRW_STAT] should be read for error information. If any bits are set in this parameter, an error occurred. This is a read-only parameter, with the following bit settings:

- Bits [7:3] = Reserved
- Bit [2] = Reserved
- Bit [1] = A mode register write was requested for the PASR mode registers (MR16, MR17) when the memory controller is not programmed for LPDDR2 memory systems (DDR_CR00[DRAM_CLASS] is not set to 'b0101).
- Bit [0] = DDR_CR45[WRMD] programming error. This bit will be set if no mode register write type is specified (DDR_CR45[WRMD] bits [23:16]) but the mode register write was triggered (DDR_CR45[WRMD] bit [25] = 'b1).
- ZQ Calibration
- Hardware Frequency Changes (MRW to MR0, MR1 and MR2 in order for each memory)
- Power-on initialization
- Power-on MR0 for initialization of non-LPDDR2 memories
- Power-on DLL reset

34.6.9.4 Mode Register Storage in the Memory Controller

There are specific parameters in the register map to hold the date to write into the mode registers. The parameters that are not listed as read-only may be written at any time, and will be programmed into the memory mode registers when the DDR_CR45[WRMD] is programmed with the specified settings.

Table 34-217. Mode Register Parameters

Memory Controller Parameter	*write_modereg* Associated Mode Write Type Bit	Memory System Correlations	
MR0_DAFN_X	Bit [16] = 'b1 Bit [17] = 'b1	• LPDDR2 (S2): • LPDDR2 (S4): • DDR3:	MR0 (read-only) MR0 (read-only) MR0
MR1_DAFN_X	Bit [16] = 'b1 Bit [17] = 'b1	• LPDDR2 (S2): • LPDDR2 (S4): • DDR3:	MR1 MR1 MR1
MR2_DAFN_X	Bit [16] = 'b1 Bit [17] = 'b1	• LPDDR2 (S2): • LPDDR2 (S4): • DDR3:	MR2 MR2 MR2
MR3_DAX	Bit [16] = 'b1 Bit [17] = 'b1	• LPDDR2 (S2): • LPDDR2 (S4): • DDR3:	MR3 MR3 MR3
MR8_DAX	Read Only	• LPDDR2 (S2): • LPDDR2 (S4): • DDR3:	MR8 (read-only) MR8 (read-only) -[1]
MR16_DAX	Bit [16] = 'b1 Bit [18] = 'b1	• LPDDR2 (S2): • LPDDR2 (S4): • DDR3:	MR16 MR16 -[2]
MR17_DAX	Bit [16] = 'b1 Bit [18] = 'b1	• LPDDR2 (S2): • LPDDR2 (S4): • DDR3:	-[3] MR17 -[4]
MR_SINDAX	Bit [23] = 'b1	-	

1. *There is no corresponding mode register for this memory system. Any attempt to write this mode register for this memory system will be ignored.*
2. *There is no corresponding mode register for this memory system. Any attempt to write this mode register for this memory system will be ignored.*
3. *This mode register should not be written for this memory system.*
4. *There is no corresponding mode register for this memory system. Any attempt to write this mode register for this memory system will be ignored.*

34.6.10 Refresh Per Command Timing

The *tref_fN* parameter is used to define the number of DRAM cycles allowed between refresh commands from the memory controller. This parameter sets the average interval between refreshes.

The actual interval may vary by up to 8 cycles from refresh to refresh, depending on other activities that are occurring in the controller at the time.

Over an infinite time period, the average interval will be equal to the number of clocks set by this parameter; however, if the user sets this parameter to be exactly equal to the specification value tREFi, then local variations might mean that the memory tREF value may be violated by a very small amount.

The amount of this violation would be 8tCK/tREF. For example, for a 400 MHz memory and tREF=64 ms, the amount of the violation would be 2.5 ns/64 ms = 0.000004%.

34.6.11 Mobile Memories DQS

For these memories, all accesses to memory must be memory burst-aligned in both address and length. The Memory Controller still processes all transactions even if the starting and/or ending address is unaligned. Therefore, for a x4 memory, a write that begins or ends on an unaligned address will result in memory corruption. Such an access may also result in taking more data from the write FIFO than what correlates to the current transaction.

34.6.12 Refresh Per Chip Select

While memory refresh operations are necessary for DRAM data integrity, they unfortunately draw a significant current and temporarily disable the memory from accepting other commands. A standard practice is to issue refresh commands to all chip selects simultaneously at each tref interval. This minimizes the downtime before another data command can be processed (trfc) but does increase peak current while the refresh is in progress.

As an alternative, the Memory Controller includes the option to individually refresh each chip select in a sequential manner. When using this feature, a memory refresh may be initiated from the user interface by setting the a refresh parameter to 'b1 or through an internal source such as the expiration of the tref_f0 counter or an exit from Self-Refresh.

Regardless of the source, when the refresh command is received, all transactions will be inhibited and a refresh sequence will be initiated. This refresh sequence issues refresh commands to one chip select at a time, moving through all of the chip selects in succession. A programmable delay (the tref_interval bit) sets the time period between when refresh commands are issued to different chip selects.

A related memory-specific bit (DDR_CR26[TRFC_F0]) sets the time that a DRAM memory must wait between receiving a refresh command to when a data transaction can be processed. When each refresh command is issued, the trfc counter inside the MC core begins a countdown from the programmed value (in the TRFC_F0 bit). With each refresh command, the counter is reset and the memory controller will only accept commands TRFC_F0 cycles after when the final refresh command has been issued. Once the TRFC _F0 time expires, normal read/write traffic and behavior resumes in the memory controller. All other transactions will be inhibited during this refresh sequence until all chip selects have been issued their refresh commands and the associated timing requirements have been met.

34.6.12.1 Programming of the *tref_f0* bits

The *tref_f0* bits sets the average interval between refreshes. The actual interval may vary by up to 8 cycles from refresh to refresh, depending on other activities that are occurring in the controller at the time. Over an infinite time period, the average interval will be equal to the number of clocks set by this bit; however, if the user sets this bit to be exactly equal to the specification value tREFi, then local variations might mean that the memory tREF value may be violated by a very small amount.

The amount of this violation would be 8tCK/tREF. For example, for a 400 MHz memory and tREF=64 ms, the amount of the violation would be 2.5 ns/64 ms = 0.000004%.

34.6.12.2 Programming of the tref_INT bits

The interval is a critical value in achieving real current savings. The following considerations should be taken for programming the DDR_CR28[TREF_INT] bits.

- A larger interval will increase the time that the memory controller is held off from processing data transactions.
- Peak current restrictions will guide the interval value. The interval should be defined to the minimum value that still meets peak current restrictions.
- Programming the interval to zero will disable the refresh per chip select logic and refresh commands will be issued simultaneously to all chip selects.

- The refresh per chip select logic requires at least one dead cycle between refresh commands, and therefore the minimum interval used is actually 4 cycles. Programming the TREF_INT bits to 1, 2 or 3 will default to 4 cycles.
- The Memory Controller runs at half the frequency as the memories. Therefore, if the TREF_INT bits is programmed to an odd number of clocks, the actual interval used will be the one clock longer than programmed
- Other factors in the design also affect timing, and therefore the actual delay between refresh commands may exceed the programmed value. This is particularly likely when the interval value programmed is very low, such as 'b004.
- When the t_{RFC} counter completes its countdown, the memory controller will accept any pending commands. Since no read or write commands may occur during the entire refresh sequence, the TREF_INT bits should always be set to a value lower than the TRFC_F0 bits.

Figure 34-196 shows a refresh sequence for a system with four chip selects and a TREF_INT of four clocks ('b100).

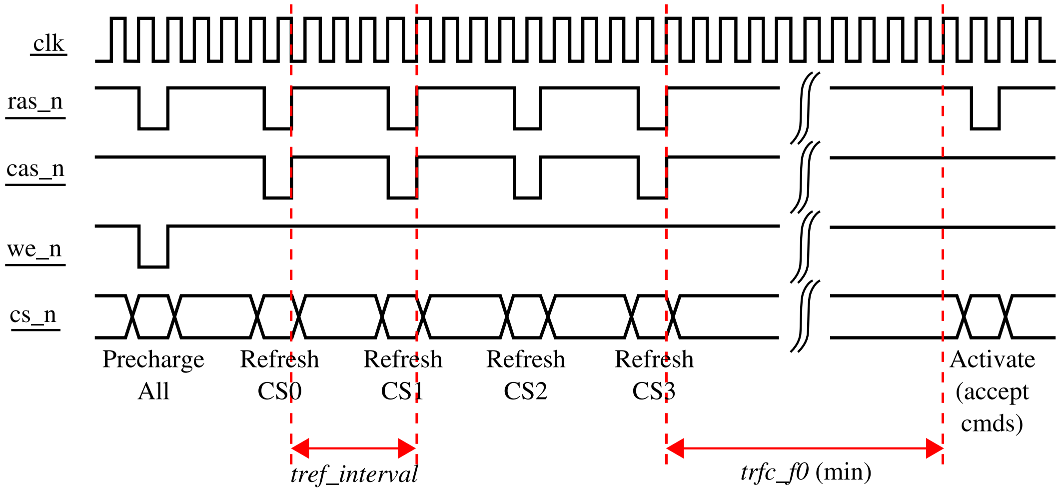


Figure 34-196. Refresh per CS Example

34.6.13 Half Data path option

The memory controller can be made to operate in half datapath mode (8 bit memory mode) through the register space with the option to enable ECC.

34.6.14 ZQ pad calibration

ZQ pad calibration logic handles the calibration of the drivers and termination resistances in DDR pads. The auto mode is handled by the logic on it's own and is the mode that would be used during functional operation of DDR. Software calibration can be used to override the pad calibration process. Software calibration is very slow and at best is useful for debug purposes.

When ZQ pad calibratrion is enabled in the DDR_CR154[PAD_ZQ_MODE], the calibration is done in parallel to the memory ZQ calibration. Pad calibration will either be done with both short and long ZQ calibration commands issued to the memory or only when long calibration is done (including during memory initial sequence and refresh exit). Memory controller must be separately programmed to control how it issues the ZQ calibration commands to the external memory.

There is provision to issue a separate one time hardware calibration .

34.6.15 DDR PHY

The DDR PHY encapsulates all functionality required to interface to external DDR DRAM devices into a single module. This module is used to control the off-chip data capture and synchronization logic for the read data. This module performs the following functions:

- Contains all data registers used to launch data, address, and control signals to the DDR memory and the memory controller.
- Controls the off-chip data capture and synchronization logic for the read data.
- Includes DLL for timing.

34.6.15.1 High Level Block Diagram

DRAM MC uses a slice-based approach for the DDR PHY. Each slice manages a byte (8 bits) of data and its corresponding DQS and DM signals.

A high level block diagram of the PHY is provided in Figure 34-197.

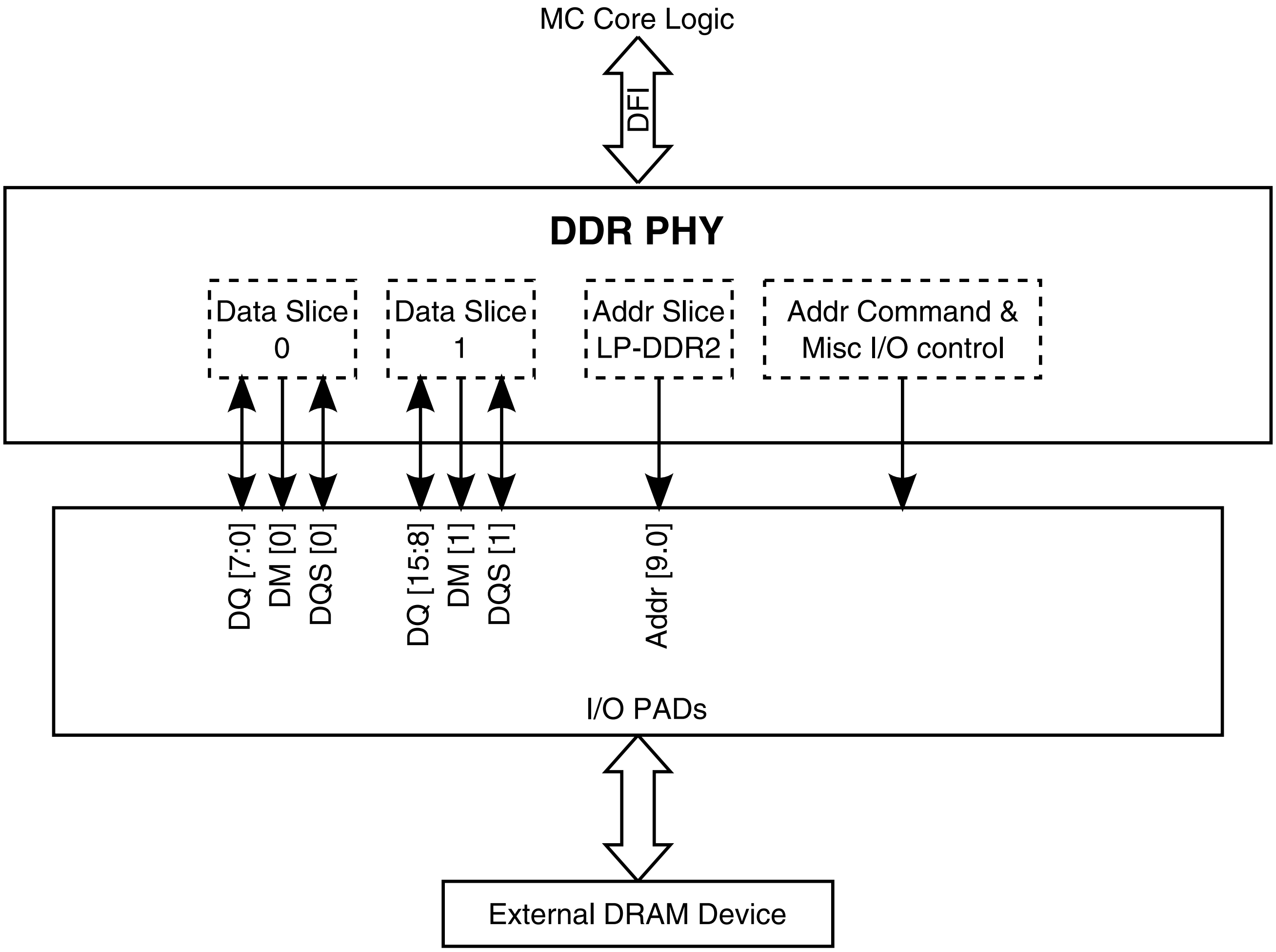


Figure 34-197. DDR PHY High Level Block Diagram

34.6.15.2 DFI

The interface between the DRAM MC core logic and the DDR_PHY is called "DFI". The "D" stands for "DRAM Controller" while "FI" is a reference to "PHY".

34.6.15.3 The I/O Timing of Address & Command

Figure 34-198 illustrates the I/O timing of Address and Command.

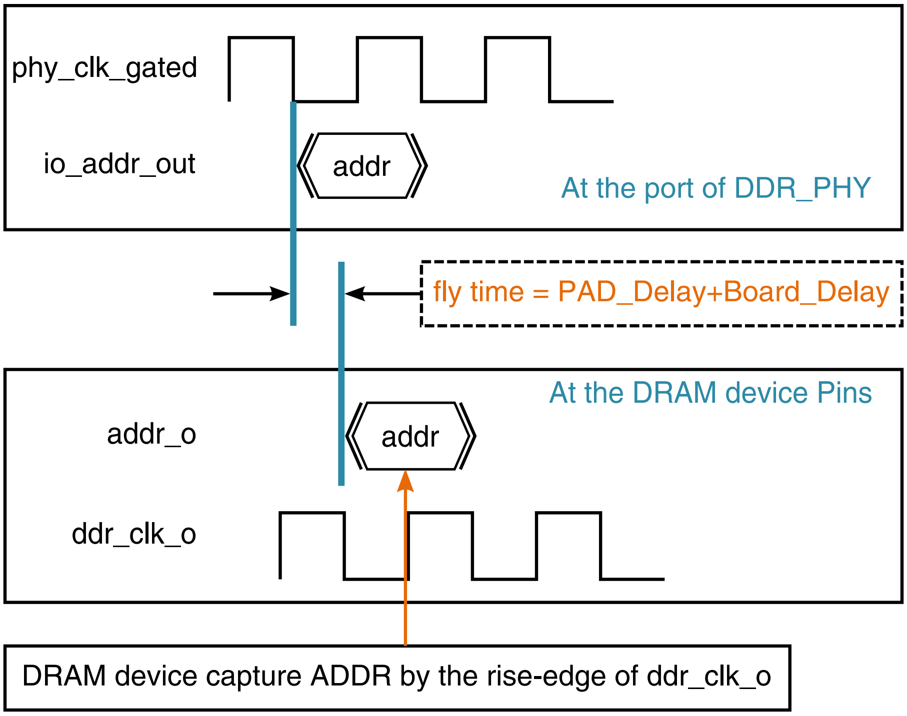


Figure 34-198. The I/O Timing of Address

The address & command signals including all DRAM interface signals except the DQ & DQS: clock-enable (CKE), chip-select, bank-address, row/cloumn_address, CASn, RASn, WEn, etc.

1. The Address & command signals is latched out by the fall-edge of PHY_CLK; In another word, the rise-edge of PHY_CLK is center-aligned with the output Address & Command signals.
2. The "fly-time" means the signals propagation time start from the DDR_PHY logic, go accross the output I/O (PAD), go through the signal trace on the board, then arrive the PIN of the DRAM device.
3. The DRAM device would capture the Address & Command signals by the rise edge of ddr_clk_o.
4. Figure 34-198 illustrates the address timing for DDR3. The timing of LPDDR2 is quite different.

34.6.15.4 The Address Timing of LPDDR2

There's a special slice "Addr Slice", illustrated in Figure 34-197. It is designed to deal with the address[9:0] of LPDDR2 device which require double-data-rate timing. Either the address[9:0] (in LPDDR2 mode) or the DDR write data timing can be illustrated by Figure 34-202.

34.6.15.5 Data Slice Overview

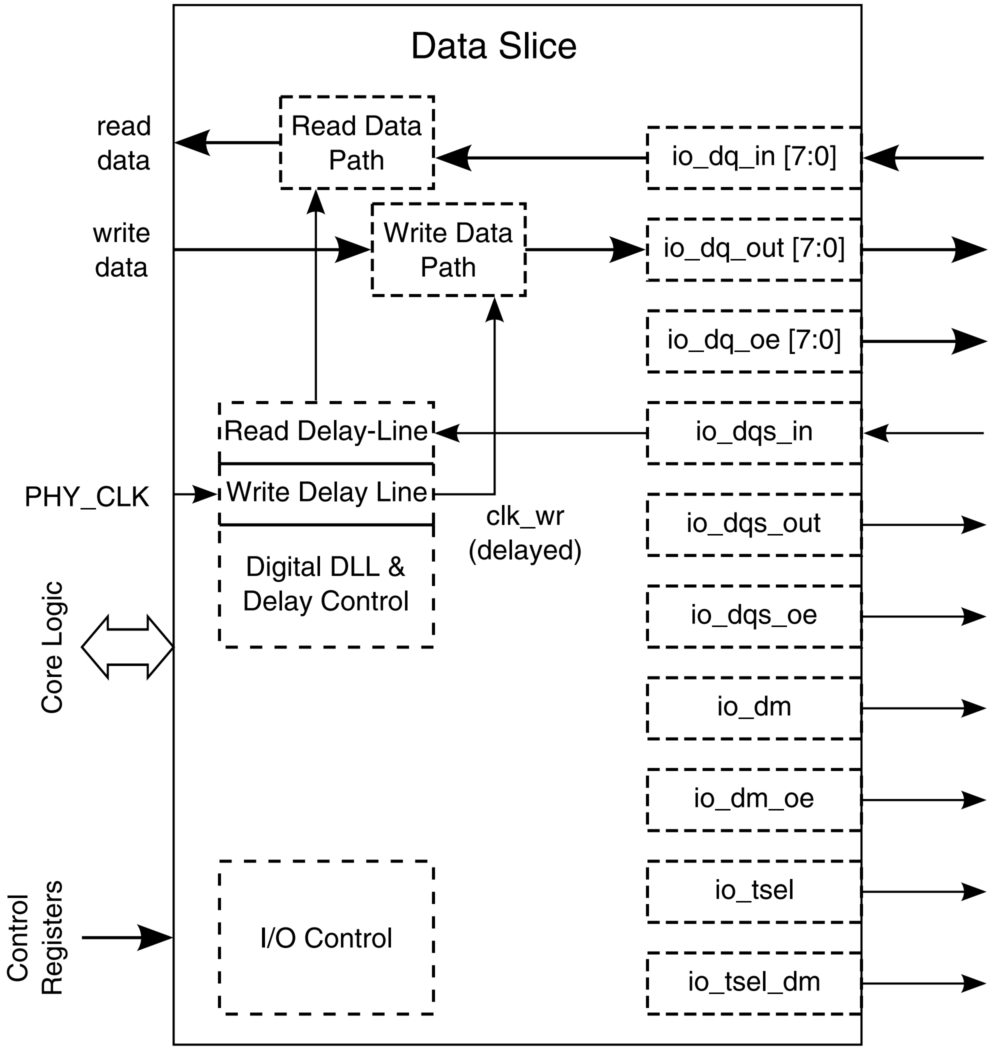


Figure 34-199. DDR PHY Data Slice

Each data-slice manages a byte (8-bit) of data and its corresponding signals.

The "Read Data Path" capture read data by latching it into read data buffers by both the posedge & negedge of "delayed_dqs". Then, the read data is synchronized from "delayed_dqs" clock domain to PHY_CLK domain.

The "Write Data Path" synchronize write data and write data mask (the "dm") from PHY_CLK domain to "dqs_out" domain.

The "Digital DLL" controls the delay values of "read delay-line" and "write delay-line". It can be bypassed and switched to manual delay control mode. In manual delay control mode, the delay values of read/write delay-line can be programmed into it's control registers separately.

34.6.15.6 Read Data Capture

When reading, DDR (dual data rate) devices send a data strobe (DQS) signal coincident with the read data. The edges of this DQS strobe are aligned (edge-aligned) with the data output by the DDR devices.

To latch read data into DRAM MC read data buffer, it requires that the latch clock edge is center-aligned with the data. Thus, a delayed version of the DQS strobe signal must be used to capture the data. Because the frequency of the DQS strobe signal is matched to the PHY_CLK, the delay is a relative number based on the period of the PHY_CLK. In the example shown in Figure 34-200, the delay is set to approximately 25% of the PHY_CLK cycle.

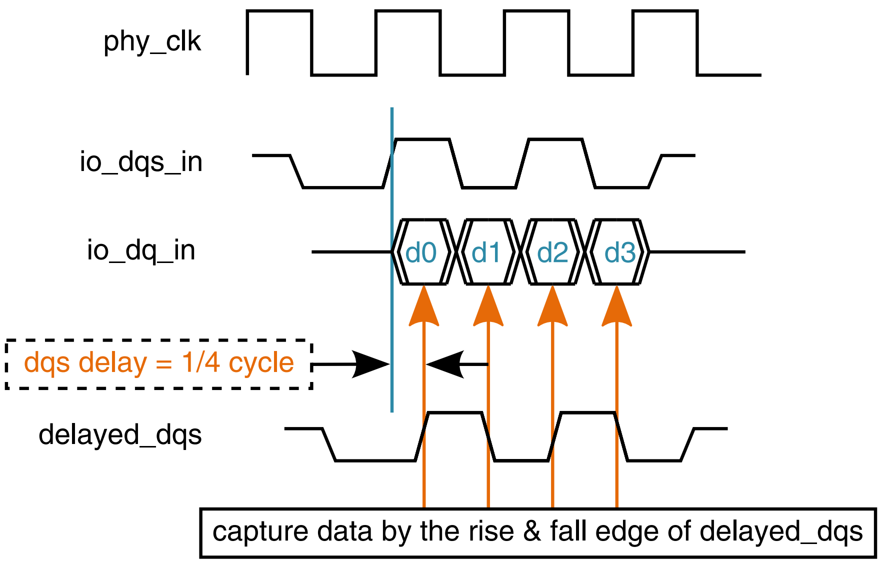


Figure 34-200. Read Data Capture

34.6.15.7 Synchronize Read Data From delayed_dqs To PHY_CLK domain

Read data is captured into data buffers by the "delayed_dqs". It must be synchronized into PHY_CLK domain, then returned to DRAM MC core logic.

Figure 34-201 illustrates how the read data from the external DRAM device is captured and synchronized to DRAM MC core logic.

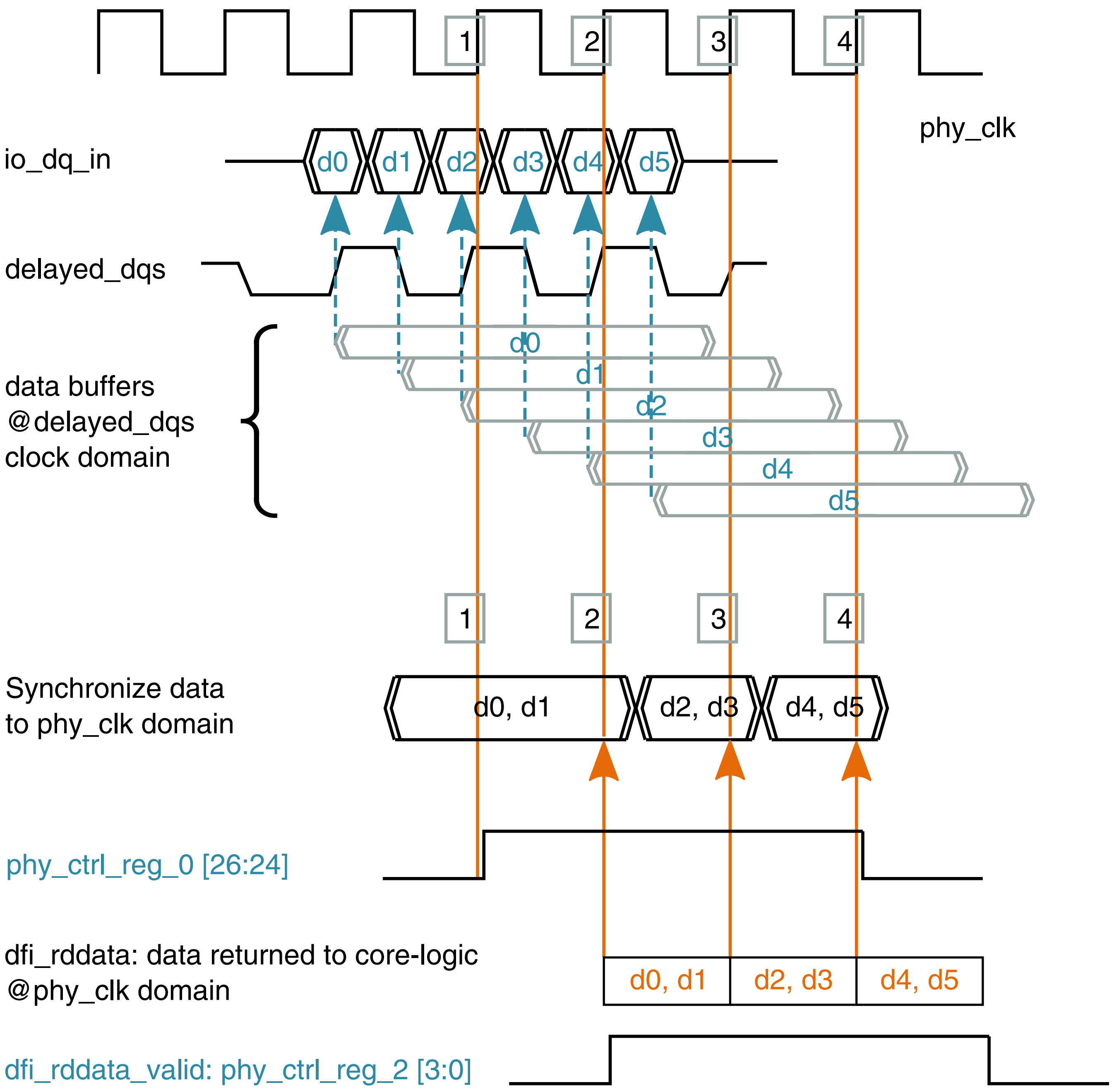


Figure 34-201. Synchronize Read Data

1. io_dq_in & io_dqs_in are not aligned with PHY_CLK in phase. Many factors affect the phase of io_dq/dqs_in, such as voltage, temperature, board layout, manufacture process, and so on.

2. Must capture the io_dq_in by delayed_dqs to meet the critical timing requirement in high frequency.
3. The data path width @PHY_CLK domain is twice of the data path width @delayed_dqs domain.
4. The DRAM MC core logic fetch read data by the rise-edge of PHY_CLK. The DRAM MC core logic fetch read data by two important signals: dfi_rddata & dfi_rddata_valid.
5. The user can program the timing position of "dfi_rddata & dfi_rddata_valid" by programmable registers "phy_ctrl_reg_0[26:24]" & "phy_ctrl_reg_2[3:0]". The unit is one cycle of PHY_CLK.
6. In this example, read data "d0,d1" getting valid before edge number "1" of PHY_CLK and is synchronized at edge "2" of PHY_CLK. Both the setup & hold time requirement are met which means read data could be fetched by core logic safely. Alternatively, read data can also be returned to core logic at edge number "1" of PHY_CLK, a cycle prior to the example. In this scenario, need to set the value of register "phy_ctrl_reg_0[26:24]" & "phy_ctrl_reg_2[3:0]" one number less. The, the "dfi_rddata" & "dfi_rddata_valid" would getting valid one cycle earlier. The benefit is that the read latency was shorten by one cycle, it help increasing the system performance. But, there's a timing risk on the "setup time", if the io_dq/dqs_in comes a little late, "setup time violation" may occur and the un-expected data would be returned to core logic, and, might lead to system crash.

34.6.15.8 Write Data Path

Figure 34-202 illustrates the write data path.

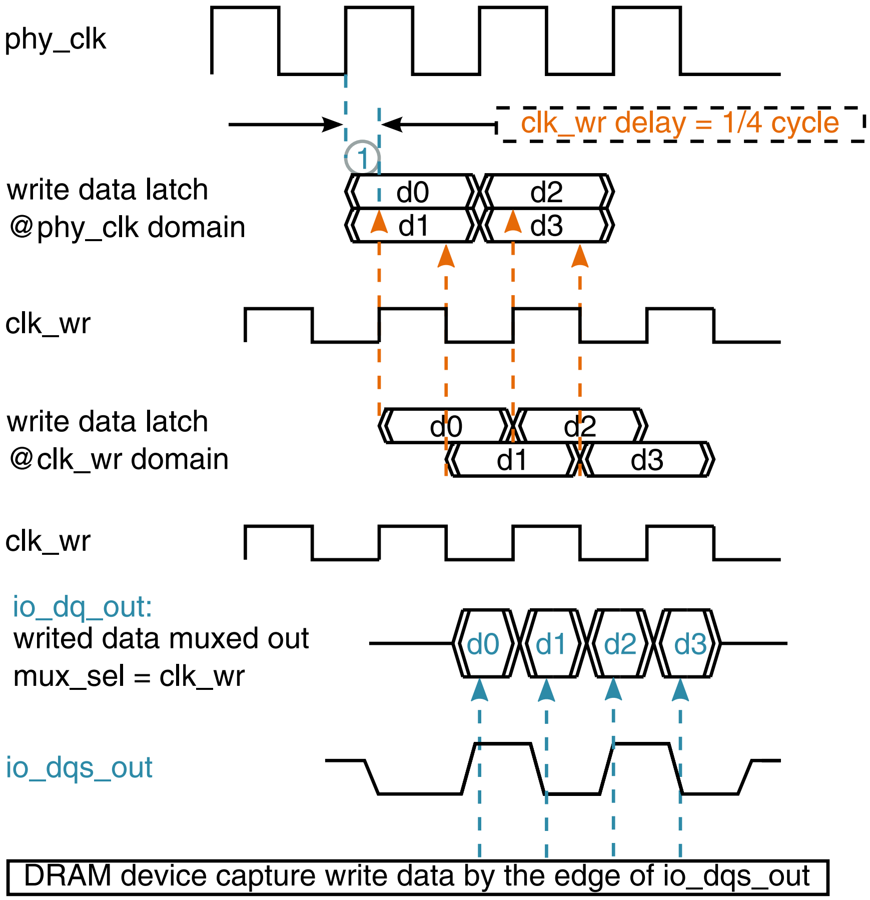


Figure 34-202. Write Data Path

NOTE

The marker "1" indicates that the "setup time" here for data "d0" is only "1/4 cycle", that would be critical in timing. The Low-Latency Option of Write Data Path , contains more detail on it.

The "io_dqs_out" and the PHY_CLK are edge-aligned in this DRAM MC design. This helps to meet the "tDQSS" timing requirement defined by the DRAM device.

The "io_dqs_out" and the "io_dq_out" are center-aligned, as required by the DRAM device.

34.6.15.8.1 The Low-Latency Option of Write Data Path

DDR_PHY provides 2 write data path options for the user :

1. The "Standard-Latency" option.

An extra latch at PHY_CLK domain is asserted into the write data path, right ahead of the latch at clk_wr domain (at the place of marker "1" in Figure 34-202). By this means, the asserted-latch and the latch@clk_wr are put back-to-back. The total delay in write data path is "1+1/4" cycle. It is safe in timing but decrease the performance by adding one cycle latency.

2. The "Low-Latency" option.

   The asserted latch for standard-latency is bypassed. The total delay in write data path is "1/4" cycle. There's risk in timing of write data path but it has higher performance than standard-latency case.

34.6.15.9 Digital DLL and the Delay-Line

Due to the asynchronous nature of the DRAM devices, the timing requirements for capturing and receiving data between the MC and the DRAM devices must be addressed. This DRAM MC contains a circuit that, in conjunction with I/O cell circuitry, can be used to meet the timing requirements for DRAM devices. The delay compensation circuit was designed with the following features:

- Programmable read strobe delay specified as a percentage of a clock cycle.
- Programmable write data delays specified as percentages of a clock cycle.
- Delay compensation circuit re-sync circuitry activated during refresh cycles to compensate for temperature and voltage drift.
- Separate delay chains for each read DQS signal from the DRAM devices.

The delay compensation circuitry relies on a master/slave approach. There is a master delay line which is used to determine how many delay elements constitute a complete cycle. This count is used, along with the programmable fractional delay settings, to determine the actual number of delay elements to program into the slave delay lines. The master and slave delay lines are identical. This approach allows the memory controller to observe a clock and then delay other signals a fixed percentage of that clock. The DLL logic does not actively generate clock signals.

The actual delay element for the delay lines is user-selectable.

Figure 34-203 shows the block diagram for a digital DLL.

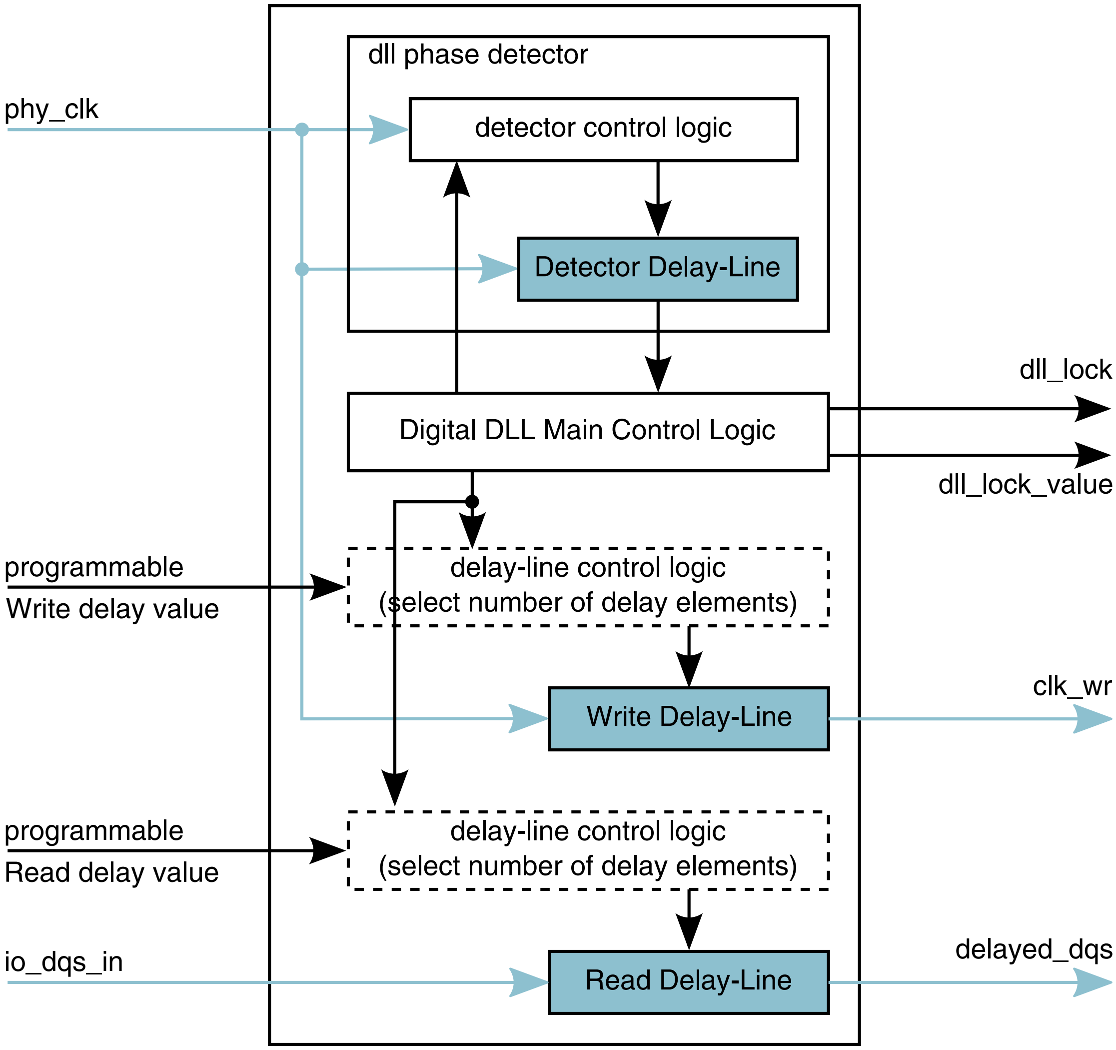


Figure 34-203. Digital DLL

The delay-line is comprised by 255 tiny delay-elements. The delay value of each delay element are almost the same. And the number of delay elements which is in use to form a delay result is configurable.

There are two modes for the delay-line control :

1. Auto-configure mode:

   This process begins by the "phase-detector". Once the phase-detector successfully complete the detection, it will raise the "dll_lock" signal and output "dll_lock_value" which indicates the delay of cone cycle. The number of elements that are needed to capture an entire clock cycle is then converted into an unsigned integer named encoder [7:0]. This integer is used as the dividend for the read and write delay parameters. The actual delay setting for the delay lines is calculated by multiplying

the encoder [7:0] integer by the parameter settings for each delay line and then dividing by 128 and rounding. These values are then encoded into a one-hot counter and updated at initialization and at every refresh interval.

2. Bypass mode, or manual configure mode:

   The phase detector can be bypassed. The number of delay elements for read & write delay line is programmed manually.

NOTE

In case the number of delay elements is fixed, the actual delay value (in nanoseconds) of a delay-line would change according to environmental conditions such as voltage, temperature, etc.

Using the auto-configure mode helps to eliminate the change of delay value because the reference delay itself also changes along with environmental conditions.

34.6.15.10 Configure the "output enable" of I/O Control

Some DRAM MC I/Os require a control signal named "output enable". Only when the "output enable" signal is active, the I/O can drive signals out, refer to Figure 34-204.

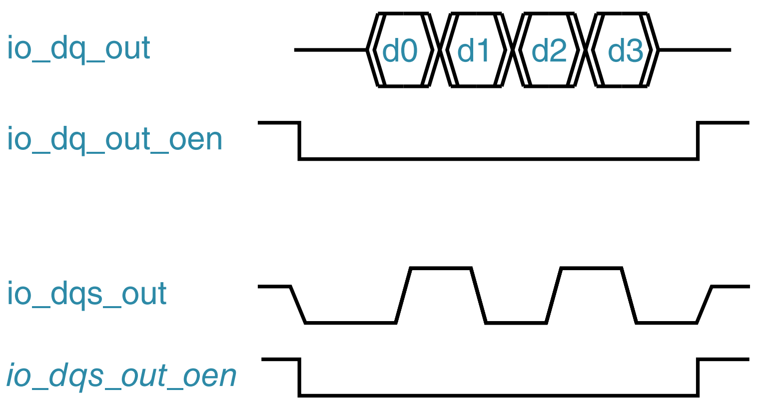


Figure 34-204. Output Enable of DRAM MC i/Os

There are 3 kinds of output enable signals in this DRAM MC design:

- io_dq_out_oen

  output enable for write data.

- io_dqm_oen

output enable for write data mask. Please configure it exactly the same as io_dq_out_oen.

- io_dqs_oen

  output enable for write data strobe.

The suffix "_oen" means the output enable signal is active low. The start time and the end time of a output enable signal can be configured separately. The start time and the end time can be adusted by 1/4 clock cycle in unit.

To meet the timing requirement of the I/O circuits, the start time of each output enable must be prior to it's corresponding output signal, and, the end time must be later than it's corresponding signal. In another word, the output enable signal must be "wider" than the output signal. The margin at start time is named "pre-amble" and the margin at end time is named "post-amble". In the example of Figure 34-204, pre-amble = 1/2 cycle while post-amble = 1/4 cycle.

34.6.16 Levelling Operations through Software

DDR3 memories feature leveling capabilities that allow more accurate alignment of critical timing signals for read and write operations. DDR3 PHYs must account for the delays induced by the fly-by topology to properly align the write DQS with the memory clock at the memory interface and to properly time the gate signal for read data capture. There are three different leveling delays which may be tuned in the DDRMC in DDR3 mode:

- Write leveling: Used to locate the delay at which the write DQS rising edge aligns with the rising edge of the memory clock.
- Gate training: Used to correctly time the opening of the gate for the read DQS.
- Read leveling: Used to center the read DQS in the DQ data eye on a memory read. This process assumes that gate training has been completed.

The DDRMC supports all of the leveling operation modes through software. Software write leveling refers to a mode of operation where software must handle the leveling sequences including setup.

34.6.16.1 Detailed Software Leveling Procedure

Leveling operations should be performed after memory initialization. In addition, a leveling operation may be initiated by software at any point, or as a response to a hardware interrupt.The DDRMC will inform the user interface of a leveling request by

hardware through the leveling request interrupt CR80[INT_STAT]. This bit is tied to the CR94[LVL_STAT], which contains a single bit per type of leveling operation and is set whenever a leveling request is sent from the PHY, if the periodic refresh timers for a leveling request are triggered, or if the request parameter is set.

Software is not required to use the CR94[LVL_STAT] or respond to a hardware interrupt for leveling requests. However, if the user wishes to count refreshes between leveling operations, then this bit will inform the user when the refresh counter expires. If desired, software may respond to a leveling interrupt by reading the CR94[LVL_STAT] and then programming the CR93[SW_LVL_MODE] with the type of leveling procedure requested. If the interrupt and lvl_status bit are not being used, software must program the CR93[SW_LVL_MODE] with the operation being implemented. The encoding of the 2-bit CR93[SW_LVL_MODE] is:

- ’b00: No leveling
- ’b01: Write leveling
- ’b10: Read leveling
- ’b11: Gate training

After specifying the type of leveling, the software algorithm should initiate the procedure by setting the CR90[SWLVL_START]to ’b1. This triggers the memory controller (MC) to complete any commands in progress, inhibit the command queue from accepting additional requests, and issue a precharge_all command to the memory to close all banks (if necessary). The memory controller will then issue the appropriate mode register commands to place the memories into write leveling mode or to enable read leveling through the MPR register. After meeting any applicable timing parameters, the MC will also assert one of the DFI leveling enable signals which enables the appropriate logic in the PHY.

The CR94[SWLVL_OP_DONE] is used to inform the software of the status of any active operations. This parameter will be cleared to ’b0 during any initialization, load or exit operation is being performed and set to ’b1 when the operation is complete. The leveling operation complete interrupt (bit 18) will also be set to ’b1 in the CR80[INT_STAT] when the CR94[SWLVL_OP_DONE] is set. The software may wait for the interrupt or monitor this parameter to indicate that the operation in process has completed, and another action may be performed. The interrupt may be disabled by setting the associated bit in the CR82[INT_MASK] to ’b1 if it is preferable not to have the interrupt firing.

The procedures for write and read leveling are influenced by the type of write leveling, gate training and read leveling support provided by the PHY and defined through the MC input signals.The procedures described in this document are only applicable for the MC Evaluation mode.

If the PHY is operating in MC Evaluation mode, after initialization, the appropriate delay parameters should be written with the value of delay that is needed for each data slice X in the PHY. The delay parameters used by the software leveling option are:

- Write Leveling: WRLVL_DLL_X bits
- Gate Training: RDLVL_GTDL_X bits
- Read Leveling: WRLVL_DL_X bits

At this point, the software should set the CR93[SWLVL_LOAD] to ’b1. This action will trigger the MC to de-assert the CR94[SWLVL_OP_DONE]and the MC will initiate a loading of the delay values into the data slices. The MC will then wait for the load operation to complete, and initiate a write level strobe or a read burst. The MC will wait for the response from the memory and save the response into the CR94[SWLVL_RESP_2], CR95[SWLVL_RESP_1],CR94[SWLVL_RESP_0] for each slice X. Once all responses are saved, the MC will assert the CR94[SWLVL_OP_DONE] and generate an interrupt. This informs the user that the response data is available for the initial delay values.

The software should use the response values CR94[SWLVL_RESP_2], CR95[SWLVL_RESP_1],CR94[SWLVL_RESP_0] to determine the next operation. For MC Evaluation mode, the response will indicate if the delay is appropriate, or must be increased or decreased. If the delay must be changed, the new delay value should be written to the associated parameter and the load sequence should be performed again.

If the delays are accurate (MC Evaluation mode), the software should now exit leveling by setting the CR94[SWLVL_EXIT] to ’b1. This triggers the MC to clear the mode registers in the memories,de-assert the DFI leveling enable signal and enable the command queue for normal operation. The software is not required to complete the entire procedure prior to exiting. The user may exit after initialization or after a load operation, even if the response does not indicate completion or accurate values. Any writes to the CR94[SWLVL_EXIT] while the CR94[SWLVL_OP_DONE] is clear will be ignored. As with other operations, the CR94[SWLVL_OP_DONE] will be de-asserted at the start of the exiting process and asserted with an interrupt generation when exit has completed.

The value in the CR93[SW_LVL_MODE] is sampled when the CR93[SWLVL_START] is asserted, and must remain at the same value until the CR94[SWLVL_EXIT] operation has been completed. If this parameter is changed during any load operations, the MC may exhibit unstable behavior.

The software leveling logic may be used to level any chip select in the system. The user may identify a particular chip select by programming the CR95[WRLVL_CS] (for write leveling) or CR101[RDLVL_CS] (for gate training or read leveling). If there are multiple chip selects in the system that must be leveled, the entire leveling procedure must be completed for one chip, then the CR95[WRLVL_CS] or CR101[RDLVL_CS] should be changed and the entire leveling procedure performed again. Only one set of delays may

be stored in the DDRMC. The DDRMC does not contain storage for the delays for multiple chip selects. If this is required for leveling, the PHY must also contain the register space to store those delay values.

NOTE

Although an interrupt is generated at the end of each step in the leveling process, the system does not have access to the memory served by the controller. The command queue is disabled during this time and the memory is in a state in which it will not accept read or write commands. Any interrupt-servicing routine will need to be developed with the knowledge that it will not have access to this memory, or the user may first mask interrupt(s) by setting the associated bits in the CR82[INT_MASK] to ’b1 before performing leveling through software.

34.6.16.2 Software Write Leveling

Write leveling is a feature of DDR3 memories that is used to align the rising edge of the memory clock and the rising edge of the write data strobe relative to each other at the memory interface, accounting for system skew. Write leveling must be done per data slice to allow proper alignment of clock and DQS.

The procedure outlined in the abovementioned Detailed Software Leveling Procedure Detailed section will be followed for write leveling, gate training and read leveling. This section details the specific procedure for write leveling through software.

Several parameters in the Controller must be set to appropriate values prior to initiating a software write leveling operation. The CR96[WRLVL_EN] must be cleared to ’b0 to disable hardware write leveling and the CR95[WRLVL_CS] must be programmed with the rank of memory to be leveled. The CR93[SW_LVL_MODE] must also be programmed to ’b01 to indicate the desire to perform a write leveling operation through software.

34.6.16.2.1 Software Write Leveling in MC Evaluation Mode

If the system is relying on the MC to evaluate the responses and set the delay values, the software should perform the following sequence:

1. Software may initiate a write leveling operation at any time, or as a response to the leveling request interrupt (bit 17) being set to ’b1 in the CR80[INT_STAT]. Software is not required to respond to the hardware interrupt.

If the hardware interrupt is being used, the refresh timer may be have expired. Read the lvl_status parameter to determine the type of leveling operation being requested. If bit [0] is set, then a write leveling operation is being requested.

2. Ensure that appropriate values are programmed in the CR139[PHY_WRLV_DLL], CR139[PHY_WRLV_EN], CR139[PHY_WRLV_LOAD], CR141[PHY_WRLV_RESP], CR139[PHY_WRLV_RESPLAT], CR140[PHY_WRLV_WW], CR17[TMOD], CR96[WLDQSEN], CR96[WLMRD].
3. Program the CR95[WRLVL_CS] for write leveling.
4. Program the CR93[SW_LVL_MODE] to ’b01.
5. Set the CR93[SWLVL_START] to ’b1.

   The MC will clear the CR94[SWLVL_OP_DONE], complete the current command, inhibit the command queue and issue a precharge_all to the memory banks if necessary. A mode register write will place the memory (identified by the CR95[WRLVL_CS] into write leveling mode and, wait TMOD cycles, the PHY write level enable signal will be asserted to enable the PHY write leveling logic. After WLDQSEN cycles, the DQS strobe is sent.

6. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt (bit 18) in the CR80[INT_STAT] until the operation has completed.
7. Program delay values for each data slice X into the WRLVL_DLL_X bits.
8. Set the CR93[SWLVL_LOAD] to ’b1.

   The MC will clear the CR94[SWLVL_OP_DONE], wait PHY write level enable signal and then load the delay values onto the PHY write level gate delay signals. After PHY write level load cycles, the PHY write level load signal will be asserted. The MC will ensure that PHY write level DLL has elapsed and then initiate a write level strobe. At PHY clocks after a write command cycles, the MC will load the responses from the PHY_WRLVL_RESP signals into the SWLVL_RESP_X bits for each data slice X.

9. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt (bit 18) in the CR80[INT_STAT] until the operation has completed.
10. Read the responses from the SWLVL_RESP_X bits.

    A “1” indicates the rising edge of the strobe occurred during the high phase of the clock and a “0” indicates the rising edge of the strobe occurred during the low phase of the clock. In general, the user will want to increase the delay if a “0” response was received or decrease the delay if a “1” response was received.

11. Based on the response for each slice, a higher or lower delay value should be programmed into each WRLVL_DLL_X bits. Note that the software must track the “previous” value of the response for future evaluation
12. Set the CR93[SWLVL_LOAD] to ’b1.

The MC will clear the CR94[SWLVL_OP_DONE], load the updated delay values onto the PHY write level delay signals, wait PHY write level load and then assert the PHY write level load signal. The MC will wait PHY write level DLL cycles and ensure that PHY WRVL_RR has elapsed before issuing another write level strobe. After PHY WRLV_RESP cycles, the MC will load the responses from the WRLV_RESP signals into the SWLVL_RESP_X bits for each data slice X.

13. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt CR80[INT_STAT] until the operation has completed
14. Read the responses from the SWLVL_RESP_X bits. The meaning of the response is based on the previous response and how the delay was changed for this iteration.
15. Repeat steps 11 through 14 until the transition point has been located, or the delay value has reached the maximum or minimum possible.
16. Set the CR94[SWLVL_EXIT] to ’b1.

    The MC will clear the CR94[SWLVL_OP_DONE] , perform a mode register write to disable the read leveling mode for the memory and de-assert the PHY write level enable signal. The command queue will be enabled for normal operation.

17. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt (bit 18) in the CR80[INT_STAT] until the operation has completed.

Table 34-218. Write Leveling Responses in MC Evaluation Mode

Previous	Current	Delay Change	Suggested Action
0	0	Incremented	Increment delay value
0	0	Decremented	Decrement delay value
0	1	Incremented	Rising edge transition point has been located.
0	1	Decremented	Decrement delay value (leveling is attempting to locate rising edge.)
1	0	Incremented	Increment delay value (leveling is attempting to locate rising edge.)
1	0	Decremented	Rising edge transition point has been located.
1	1	Incremented	Increment delay value
1	1	Decremented	Decrement delay value

34.6.16.2.2 Software Write Leveling Software Considerations

There are certain cases that the software should be prepared to consider for write leveling:

1. The delay value may reach the minimum or maximum possible without finding the edges of the clock relative to the strobe. In this case, the software should determine whether the smallest or largest value of the delay line is most appropriate, or whether another adjustment is required to place the signal within the range of the delay line.
2. If the initial delay value causes the strobe to just miss the rising edge of the clock, a full cycle of delay may be required to find the next rising edge. However, this full cycle of delay may cause the data to ultimately be delayed a cycle longer than needed. If the initial strobe just misses the clock edge, the initial value may be the correct value of delay. For example, if the initial delay tested responded with a “1”, then the correct delay may be 0x0.
3. The software must be aware of any limitations of the delay lines in the PHY, including maximum delay values that can be programmed.
4. Once the data slices have been leveled, incremental re-leveling may be accomplished more quickly by starting with the previous value and incrementing or decrementing the delay based on the responses.
5. Ideally, the signal should be placed away from the edges of the delay line to allow flexibility in adjustment.

34.6.16.3 Software Gate Training

Gate training is a feature of DDR3 memories that is used to align the rising edge of the read data strobe gate in the middle of the preamble of the read data strobe signal at the memory interface. The procedure outlined in abovementioned Detailed Software Leveling Procedure section will be followed for write leveling, gate training and read leveling. This section details the specific procedure for gate training through software.

Several parameters in the Controller must be programmed to appropriate values prior to initiating a software gate training operation. The CR149[RDLVL_GATE_EN]must be cleared to ’b0 to disable hardware gate training and the CR101[RDLVL_CS] must be programmed with the rank of memory to be leveled.

Since the gate is being aligned to the first read DQS, the user should adjust the gate by a 1/2 cycle forward prior to performing gate training. Then, the 1/2 cycle delay should be removed, aligning the gate in the center of the preamble. In addition, the DFI specification requires that the gate signals must assert during the preamble or first high phase of the read DQS.

The DFI training interface timing parameters must be defined for the system. The CR93[SW_LVL_MODE] must also be set to ’b11 to indicate the desire to perform a gate training operation through software.

34.6.16.3.1 Software Gate Training in MC Evaluation Mode

The software should perform the following sequence:

1. Software may initiate a write leveling operation at any time, or as a response to the leveling request interrupt (bit 17) being set to ’b1 in the CR80[INT_STAT]. Software is not required to respond to the hardware interrupt.

   If the hardware interrupt is being used, the refresh timer may be have expired. Read the CR90[LVL_STATUS] parameter to determine the type of leveling operation being requested. If bit [0] is set, then a write leveling operation is being requested.

2. Ensure that appropriate values are programmed in the CR144[PHY_RDRLV_DLL], CR144[PHY_RDLV_EN], CR144[PHY_RDLV_LOAD], CR144[PHY_RDLV_RES], CR146[PHY_RDLV_RESP], CR145[PHY_RDLV_RR].
3. Program the CR101[RDLVL_CS] with the target chip select for gate training.
4. Program the CR93[SW_LVL_MODE] to ’b01.
5. Set the CR93[SW_LVL_MODE] to ’b11.
6. Set the CR93[SWLVL_START] to ’b1.

   The MC will clear the CR94[SWLVL_OP_DONE], complete the current command, inhibit the command queue and issue a precharge_all to the memory banks if necessary. An MPR write will enable read leveling for the memory (identified by the CR101[RDLVL_CS) and the PHY read level gate enable signal will be asserted to enable the PHY gate training logic.

7. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt (bit 18) in the CR80[INT_STAT] until the operation has completed.
8. Program delay values for each data slice X into the RDLVL_GTDL_X bits.
9. Set the CR93[SWLVL_LOAD] to ’b1.

   The MC will clear the CR94[SWLVL_OP_DONE], wait PHY read level enable signal and then load the delay values onto the PHY read level gate delay signals. After PHY read level load cycles, the PHY read level load signal will be asserted. The MC will ensure that PHY read level DLL has elapsed and then initiate a read burst. At PHY clocks after a read command cycles, the MC will load the responses from the PHY_RDLVL_RESP signals into the SWLVL_RESP_X bits for each data slice X.

10. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt (bit 18) in the CR80[INT_STAT] until the operation has completed.

    The value returned will be the sampled value of DQS at the rising edge of the gate signal. In general, the user will want to increase the delay if a “0” response was received or decrease the delay if a “1” response was received.

11. Based on the response for each slice, a higher or lower delay value should be programmed into each RDLVL_GTDL_X bits. Note that the software must track the "previous" value of the response for future evaluation.
12. Set the CR93[SWLVL_LOAD] to 'b1. The MC will clear the CR94[SWLVL_OP_DONE], load the updated delay values onto the PHY RDLVL_GTDL_X signals, wait PHY read level load and then assert the PHY read level load signal. The MC will wait PHY read level DLL cycles and ensure that PHY RDVL_RR has elapsed before issuing another read burst. After PHY RDLVL_RESP cycles, the MC will load the responses from the RDLVL_RESP signals into the SWLVL_RESP_X bits for each data slice X.
13. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt CR80[INT_STAT] until the operation has completed.
14. Read the responses from the SWLVL_RESP_X bits. The meaning of the response is based on the previous response and how the delay was changed for this iteration.
15. Repeat steps 11 through 14 until the gate is aligned to the rising edge of the read DQS for each data slice, or the delay value has reached the maximum or minimum possible
16. Set the CR94[SWLVL_EXIT] to 'b1.

    The MC will clear the CR94[SWLVL_OP_DONE] , perform a mode register write to disable the read leveling mode for the memory and de-assert the PHY read level gate enable signal. The command queue will be enabled for normal operation.

17. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt (bit 18) in the CR80[INT_STAT] until the operation has completed.

Table 34-219. Gate Training Responses in MC Evaluation Mode

Previous	Current	Delay Change	Suggested Action
0	0	Incremented	Increment delay value
0	0	Decremented	Decrement delay value
0	1	Incremented	Rising edge transition point has been located.
0	1	Decremented	Decrement delay value
1	0	Incremented	Increment delay value
1	0	Decremented	Rising edge transition point has been located.
1	1	Incremented	Increment delay value
1	1	Decremented	Decrement delay value

34.6.16.3.2 Software Gate Training Software Considerations

There are certain cases that the software should be prepared to consider for write leveling:

1. If the delay value reaches the minimum or maximum possible without finding the edge of the DQS, an error has occurred in gate training.
2. Once the data slices have been leveled, incremental re-leveling may be accomplished more quickly by starting with the previous value and incrementing or decrementing the delay based on the responses.
3. Once gate training has been completed, the user should use read and write data patterns to verify that the gate was placed correctly within the preamble.
4. Ideally, the signal should be placed away from the edges of the delay line to allow flexibility in adjustment.

34.6.16.4 Software Read Leveling

Read leveling training is a feature of DDR3 memories that is used to align the rising edge of the read data strobe in the middle of the data eye at the memory interface. The procedure outlined in Software Leveling Procedure Detailed section, will be followed for write leveling, gate training and read leveling. This section details the specific procedure for read leveling through software.

Several parameters in the Controller must be programmed to appropriate values prior to initiating a software read leveling operation. The CR148[RDLVL_EN] must be cleared to ’b0 to disable hardware read leveling and the CR101[RDLVL_CS] must be programmed with the rank of memory to be leveled. The leveling edge must be identified in the CR101[RDLVL_EDGE]. The DFI training interface timing parameters must be defined for the system. The CR93[SW_LEVELING_MODE] must also be programmed to ’b10 to indicate the desire to perform a read leveling operation through software.

34.6.16.4.1 Software Read Leveling in MC Evaluation Mode

The software should perform the following sequence:

1. Software may initiate a write leveling operation at any time, or as a response to the leveling request interrupt (bit 17) being set to ’b1 in the CR80[INT_STAT]. Software is not required to respond to the hardware interrupt.

   If the hardware interrupt is being used, the refresh timer may be have expired. Read the CR90[LVL_STATUS] parameter to determine the type of leveling operation being requested. If bit [0] is set, then a write leveling operation is being requested.

2. Ensure that appropriate values are programmed in the CR144[PHY_RDRLV_DLL], CR144[PHY_RDLV_EN], CR144[PHY_RDLV_LOAD], CR144[PHY_RDLV_RES], CR146[PHY_RDLV_RESP], CR145[PHY_RDLV_RR].
3. Program the CR101[RDLVL_CS] with the target chip select for gate training.
4. Program the leveling edge in the CR101[RDLVL_EDGE] parameter
5. Program the CR93[SW_LVL_MODE] to ’b01.
6. Set the CR93[SWLVL_START] to ’b1.

   The MC will clear the CR94[SWLVL_OP_DONE], complete the current command, inhibit the command queue and issue a precharge_all to the memory banks if necessary. An MPR write will enable read leveling for the memory (identified by the CR101[RDLVL_CS) and the PHY read level gate enable signal will be asserted to enable the PHY read levelling logic.

7. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt (bit 18) in the CR80[INT_STAT] until the operation has completed.
8. Program delay values for each data slice X into the RDLVL_GTDL_X bits.
9. Set the CR93[SWLVL_LOAD] to ’b1.

   The MC will clear theCR94[SWLVL_OP_DONE], wait PHY read level enable signal and then load the delay values onto the PHY read level gate delay signals. After PHY read level load cycles, the PHY read level load signal will be asserted. The MC will ensure that PHY read level DLL has elapsed and then initiate a read burst. At PHY clocks after a read command cycles, the MC will load the responses from the PHY_RDLVL_RESP signals into the SWLVL_RESP_X bits for each data slice X.

10. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt (bit 18) in the int_status parameter until the operation has completed.
11. Based on the response for each slice, a higher or lower delay value should be programmed into each RDLVL_GTDL_X bits. Note that the software must track the “previous” value of the response for future evaluation.
12. Read the responses from the SWLVL_RESP_X bits. The value returned will be the value of DQ sampled by DQS for each data slice X.
13. Set the CR93[SWLVL_LOAD] to ’b1.

    The MC will clear the CR94[SWLVL_OP_DONE], load the updated delay values onto the PHY RDLVL_GTDL_X signals, wait PHY read level load and then assert the PHY read level load signal. The MC will wait PHY read level DLL cycles and ensure that PHY RDVL_RR has elapsed before issuing another read burst. After PHY RDLVL_RESP cycles, the MC will load the responses from the RDLVL_RESP signals into the SWLVL_RESP_X bits for each data slice X.

14. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt CR80[INT_STAT] until the operation has completed.
15. Read the responses from the SWLVL_RESP_X bits. The meaning of the response is based on the previous response and how the delay was changed for this iteration.
16. Repeat steps 12 through 15 until the gate is aligned to the rising edge of the read DQS for each data slice, or the delay value has reached the maximum or minimum possible.
17. The ideal delay value for this system will be the midpoint between the rising and falling edge transition points. The midpoint values should be calculated and stored in the RDLVL_DLL_X bits for each data slice X.
18. Set the CR94[SWLVL_EXIT] to ’b1.

    The MC will clear the CR94[SWLVL_OP_DONE] , perform a mode register write to disable the read leveling mode for the memory and de-assert the PHY read level gate enable signal. The command queue will be enabled for normal operation.

19. Poll the CR94[SWLVL_OP_DONE] or monitor the leveling operation complete interrupt (bit 18) in the CR80[INT_STAT] until the operation has completed.

Table 34-220. Read Leveling Responses in MC Evaluation Mode

Previous	Current	Delay Change	Suggested Action
0	0	Incremented	Increment delay value
0	0	Decremented	Decrement delay value
0	1	Incremented	Rising edge transition point has been located.
0	1	Decremented	Falling edge transition point has been located.
1	0	Incremented	Falling edge transition point has been located.
1	0	Decremented	Rising edge transition point has been located.
1	1	Incremented	Increment delay value
1	1	Decremented	Decrement delay value

34.6.16.4.2 Software Read Leveling Software Considerations

There are certain cases that the software should be prepared to consider for read leveling:

1. Depending on the resolution of the delay elements, as the DQS edge gets near the data edge, the response values may oscillate. This effect should be considered when evaluating whether a specific edge has been found. This is particularly relevant when the first delay adjustment seems to indicate an edge: several samples may be required to determine which edge has been found.

2. The software must be aware of any limitations of the delay lines in the PHY, including maximum delay values that can be programmed.
3. Once the data slices have been leveled, incremental re-leveling may be accomplished more quickly by starting with the previous value and incrementing or decrementing the delay based on the responses.
4. Ideally, the signal should be placed away from the edges of the delay line to allow flexibility in adjustment.

34.7 Initialization and Application Information

The memory controller requires a sequence for correct operation after power to the microcontroller and memory devices is stable. When power is stable, the memory controller automatically initializes the memory devices.

The procedure to initialize the memory controller is as follows:

1. Assert system reset (managed by device logic) and wait for the reset to get deasserted.
2. Issue write register commands to configure the DRAM protocols and the settings for the memory. Keep the DDR_CR00[START] de-asserted during this initialization step.
3. Assert the DDR_CR00[START]. This triggers the memory controller to execute the initialization sequence using the bits written into the registers.

The MC will set the initialization complete interrupt (bit 9) in the DDR_CR80[INT_STAT] when it is ready for commands. In addition, the user should check that the CCM_CCSR[DDRC_CLK_SEL] is enabled before accessing the DDR memory.

CAUTION

Failure to access the DDR until its clock is enabled from CCM will cause the device to hang.

Chapter 35
On-Chip One Time Programmable (OCOTP) Controller

35.1 Introduction

The OCOTP controller provides the primary mechanism for interfacing with on-chip fuses. Among the uses for the fuses are unique chip identifiers, mask revision numbers, cryptographic keys, security configuration, boot characteristics and various control signals requiring permanent non-volatility. For security purposes, the fuses protect the confidentiality or integrity of critical security data against both software attacks and board-level hardware attacks.

This chapter describes the on-chip One Time Programmable (OTP) Controller. It describes the block functionality and implementation in detail. In this document, the words "eFuse" and "OTP" are interchangeable. OCOTP refers to the hardware block itself.

35.2 Overview of On-Chip OTP (OCOTP) controller

The OCOTP controller provides the following features :

- Shadow cache of fuse values loaded at reset prior to system boot.
- Ability to read and override fuse values in shadow cache (doesn’t affect fuse element).
- Ability to read fuses directly (ignoring shadow cache).
- Ability to write (program) fuses by the software
- Fuses and shadow cache bits enforcing read-protect, override-protect, and write-protect.
- Scan protection
- Provide program-protect and read-protect efuse.
- Provide overide and read protection of shadow register.
- CRC32 test for read-lock fuse content.

35.3 Top-level symbol and functional overview

The figure below shows the OCOTP system level diagram.

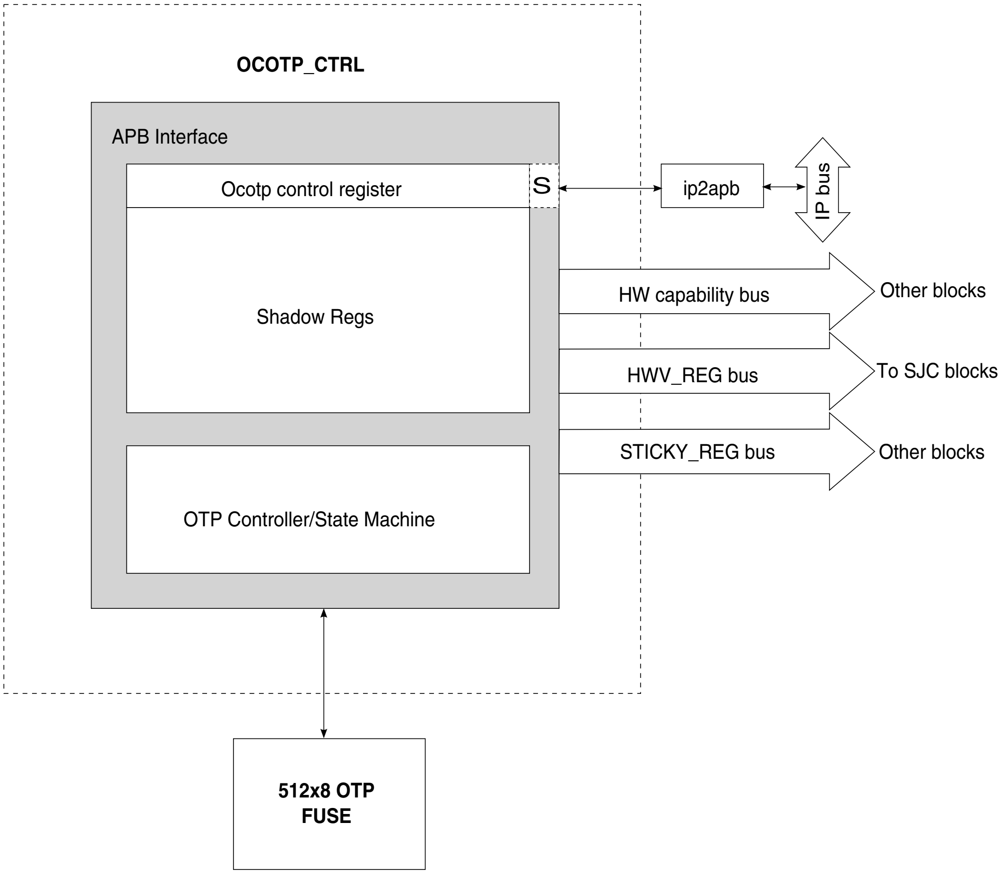


Figure 35-1. OCOTP system level diagram

35.3.1 Operation

The OCOTP provides two functions:

- Configure control registers for programming and a reading fuse word.
- Override and read shadow registers.

For an efuse, programming can only be performed on a bit and reading is based on a byte. OCOTP configuration for program and read are performed on 32-bit words for software convenience. For writes, the 32-bit word reflects the "write-mask". Bits are set to 0 by default, and when blown, are permanently set to 1. The OCOTP will program bit with 1 in the fuse word one bit by one bit. For reads, OCOTP will read four times to get 4 bytes in the fuse word in order.

This device uses nine banks of fuses. Each bank is made up of eight fuse words. Physically, there are 4k fuse bits (16 banks) present in one 512x8 efuse box. Access to the unallocated seven banks of fuses sets the ERROR bit in the OTP Controller Control Register (OCOTP_CTRL).

35.3.1.1 Fuse shadow memory footprint

The OTP memory footprint is shown in the figure below. It has 16 banks of 256 fuses (4 KB fuses in total). Each bank is arranged in 8 rows (32-bit words) with 32 fuses per row. The registers are grouped by lock region. Their names correspond to the PIO register and fusemap names.

NOTE

Out of 16 banks, 9 banks are used; remaining banks are reserved.

0x27	GP2				
0x26	GP1				
0x25	MAC1				
0x24	MAC1				
0x23	MAC2				
0x22	MAC2				
0x21	SJC				
0x20	SJC				
0x1F	RESERVED	0x47	RESERVED	0x7F	CRC7
0x1E	RESERVED	0x46	RESERVED	0x7E	CRC6
0x1D	RESERVED	0x45	RESERVED	0x7D	CRC5
0x1C	RESERVED	0x44	RESERVED	0x7C	CRC4
0x1B	RESERVED	0x43	RESERVED	0x7B	CRC3
0x1A	RESERVED	0x42	RESERVED	0x7A	CRC2
0x19	RESERVED	0x41	RESERVED	0x79	CRC1
0x18	RESERVED	0x40	RESERVED	0x78	CRC0
0x17	RESERVED	0x3F	ANALOG		.
0x16	RESERVED	0x3E	ANALOG		
0x15	RESERVED	0x3D	ANALOG		.
0x14	RESERVED	0x3C	PMU		
0x13	RESERVED	0x3B	PMU		.
0x12	RESERVED	0x3A	ANATOP		
0x11	RESERVED	0x39	RESERVED		.
0x10	RESERVED	0x38	RESERVED		
0x0F	ANALOG	0x37	RESERVED		.
0x0E	ANALOG	0x36	RESERVED		
0x0D	ANALOG	0x35	RESERVED		.
0x0C	MEM	0x34	RESERVED		
0x0B	MEM	0x33	RESERVED		.
0x0A	MEM	0x32	RESERVED		
0x09	MEM	0x31	RESERVED		.
0x08	MEM	0x30	RESERVED		
0x07	BOOT_CFG	0x2F	RESERVED	0x4F	RESERVED
0x06	BOOT_CFG	0x2E	FIELD_RETURN	0x4E	RESERVED
0x05	BOOT_CFG	0x2D	MISC_CONF	0x4D	RESERVED
0x04	RESERVED	0x2C	RSVD	0x4C	RESERVED
0x03	RESERVED	0x2B	RSVD	0x4B	RESERVED
0x02	RESERVED	0x2A	RSVD	0x4A	RESERVED
0x01	RESERVED	0x29	RSVD	0x49	RESERVED
0x00	LOCK	0x28	RESERVED	0x48	RESERVED

Figure 35-2. OTP memory footprint

35.3.1.1.1 Fuse map address table

Table 35-1. Fuse map address table

Bank	Word	Fuse Address	Base Address	Description
Bank0	Word0	0x00	0x400a5400	Lock controls
Bank0	Word1	0x01	0x400a5410	Secure JTAG Challenge
Bank0	Word2	0x02	0x400a5420	Secure JTAG Challenge
Bank0	Word5	0x04	0x400a5450	Boot Configuration
Bank1	Word7	0x0F	0x400A_54F0	General Purpose Customer Defined Information
Bank2	Word0	0x10	0x400A_5500	Reserved
Bank2	Word1	0x11	0x400A_5510	Reserved
Bank2	Word2	0x12	0x400A_5520	Reserved
Bank2	Word3	0x13	0x400A_5530	Reserved
Bank2	Word4	0x14	0x400A_5540	Reserved
Bank2	Word5	0x15	0x400A_5550	Reserved
Bank2	Word6	0x16	0x400A_5560	Reserved
Bank2	Word7	0x17	0x400A_5570	Reserved
Bank3	Word0	0x18	0x400A_5580	Reserved
Bank3	Word1	0x19	0x400A_5590	Reserved
Bank3	Word2	0x1A	0x400A_55A0	Reserved
Bank3	Word3	0x1B	0x400A_55B0	Reserved
Bank3	Word4	0x1C	0x400A_55C0	Reserved
Bank3	Word5	0x1D	0x400A_55D0	Reserved
Bank3	Word6	0x1E	0x400A_55E0	Reserved
Bank3	Word7	0x1F	0x400A_55F0	Reserved
Bank4	Word0	0x20	0x400A_5600	Secure JTAG Response Field
Bank4	Word1	0x21	0x400A_5610	Secure JTAG Response Field
Bank4	Word2	0x22	0x400A_5620	MAC Address
Bank4	Word3	0x23	0x400A_5630	MAC Address
Bank4	Word4	0x24	0x400A_5640	MAC Address
Bank4	Word5	0x25	0x400A_5650	MAC Address
Bank4	Word6	0x26	0x400A_5660	Hardware Capabilities
Bank4	Word7	0x27	0x400A_5670	Hardware Capabilities
Bank5	Word0	0x28	0x400A_5680	Reserved
Bank5	Word7	0x2F	0x400A_56F0	Hardware Capabilities

Table continues on the next page...

Table 35-1. Fuse map address table (continued)

Bank	Word	Fuse Address	Base Address	Description
Bank7	Word0	0x38	0x400A_5880	Configuration and Manufacturing Information. Bits 15:12 are allocated for ADC1 Calibration Value, Bits 11:8 are allocated for ADC0 Calibration Value.
Bank7	Word1	0x39	0x400A_5890	Configuration and Manufacturing Information. Bits 15:12 are allocated for ADC1 Calibration Value, Bits 11:8 are allocated for ADC0 Calibration Value.
Bank7	Word2	0x3A	0x400A_58A0	Configuration and Manufacturing Information. Bits 15:12 are allocated for ADC1 Calibration Value, Bits 11:8 are allocated for ADC0 Calibration Value.
Bank7	Word3	0x3B	0x400A_58B0	Configuration and Manufacturing Information. Bits 15:12 are allocated for ADC1 Calibration Value, Bits 11:8 are allocated for ADC0 Calibration Value.
Bank7	Word4	0x3C	0x400A_58C0	Configuration and Manufacturing Information. Bits 15:12 are allocated for ADC1 Calibration Value, Bits 11:8 are allocated for ADC0 Calibration Value.
Bank7	Word5	0x3D	0x400A_58D0	Configuration and Manufacturing Information. Bits 15:12 are allocated for ADC1 Calibration Value, Bits 11:8 are allocated for ADC0 Calibration Value.

Table continues on the next page...

Table 35-1. Fuse map address table (continued)

Bank	Word	Fuse Address	Base Address	Description
Bank7	Word6	0x3E	0x400A_58E0	Configuration and Manufacturing Information. Bits 15:12 are allocated for ADC1 Calibration Value, Bits 11:8 are allocated for ADC0 Calibration Value.
Bank7	Word7	0x3F	0x400A_58F0	Configuration and Manufacturing Information. Bits 15:12 are allocated for ADC1 Calibration Value, Bits 11:8 are allocated for ADC0 Calibration Value.
Bank15	Word0	0x78	0x400A_5C80	CRC0
Bank15	Word1	0x79	0x400A_5C90	CRC1
Bank15	Word2	0x7A	0x400A_5CA0	CRC2
Bank15	Word3	0x7B	0x400A_5CB0	CRC3
Bank15	Word4	0x7C	0x400A_5CC0	CRC4
Bank15	Word5	0x7D	0x400A_5CD0	CRC5
Bank15	Word6	0x7E	0x400A_5CE0	CRC6
Bank15	Word7	0x7F	0x400A_5CF0	CRC7

35.3.1.2 Fuse values

- All fuses are read and stored in a shadow register before the reset is de-asserted.
- It is possible to re-write, override these shadow register values based on the protection offered by the corresponding fuse locks.
- Fuse values can be read through shadow registers or by initiating a read sequence on the fuse location.

35.3.1.3 Fuse protection and overrides

- The first 32 bits of the fuse box define different locks. These locks are associated with rest of the fuses in the fuse box. These locks define the property of the fuses they are associated with.
- The lock bits are used to disable the fuses and the corresponding shadow registers.

- Except lock word, all the fuse words and shadow registers can be prohibited from programming or writing by programming corresponding lock bit in lock word.
- There are different types of locks defined and associated with various fuses. These fuses offer the following capabilities:
  - Override protect (OP): Fuse values cannot be overridden by writing into shadow registers
  - Write Protect (WP): Fuse writes are blocked
  - Read Protect (RP): Fuses values cannot be read by software (shadow registers).

35.3.1.4 Fuse blowing

Fuse blow is a simple procedure. Simple code is already available with Validation. Sophisticated drivers can be had from i.MX which will be taking care of the fuse and corresponding lock blow as well. It does not need any extra hardware/supply. High voltage needed to blow the fuse is already available within the device.

35.3.1.5 Fuse and shadow register read

All shadow registers are always readable through the APB bus except some secret key regions. When their corresponding fuse lock bits are set, the shadow registers also become read locked. After read locking, reading from these registers will return 0xBADABADA. In addition OCOTP_CTRL[ERROR] will be set. It must be cleared by software before any new write, read, or reload access can be issued. Subsequent reads to unlocked shadow locations will still work successfully.

To read fuse word directly from fusebox correctly, complete the following steps:

1. Program OCOTP_TIMING[STROBE_READ] and OCOTP_TIMING[RELAX] fields with timing values to match the current frequency of the ipg_clk. Refer toOTP Read/Write Timing Parameters for more information. OTP read will work at maximum bus frequencies as long as the OCOTP_TIMING parameters are set correctly.
2. Check that OCOTP_CTRL[BUSY] and OCOTP_CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read or reload must be completed before a read access can be requested.
3. Write the requested address to OCOTP_CTRL[ADDR].
4. Set OCOTP_READ_CTRL[READ_FUSE] to 1. OCOTP will auto read 4 bytes in requested word address in fusebox one by one. Then put read value into OCOTP_READ_FUSE_DATA register.
5. Once complete, the controller will clear OCOTP_CTRL[BUSY]. A read request to a protected or locked region will result in no OTP access and no setting of

OCOTP_CTRL[BUSY]. In addition OCOTP_CTRL[ERROR] will be set. It must be cleared by software before any new access can be issued.
6. Read the OCOTP_READ_FUSE_DATA register to get fuse word value. OCOTP_READ_FUSE_DATA will be 0xBADABADA when OCOTP_CTRL[ERROR] is set.

35.3.1.6 Fuse and Shadow Register Writes

Shadow register bits can be overridden by software until the corresponding fuse lock bit for the region is set. When the lock shadow bit is set, the shadow registers for that lock region become write locked. The OCOTP_LOCK register also has no shadow or fuse lock bits but it is always read only.

In order to avoid "rogue" code performing erroneous writes to OTP, a special unlocking sequence is required for writes to the fuse banks. To program fuse bank correctly complete the following steps:

1. Program OCOTP_TIMING[STROBE_PROG] and OCOTP_TIMING[RELAX] fields with timing values to match the current frequency of the ipg_clk. OTP writes will work at maximum bus frequencies as long as the OCOTP_TIMING parameters are set correctly.
2. Check that OCOTP_CTRL[BUSY] and OCOTP_CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write or reload must be completed before a write access can be requested.
3. Write the requested fuse address to OCOTP_CTRL[ADDR] and program the unlock code into OCOTP_CTRL[WR_UNLOCK]. This must be programmed for each write access. The lock code is documented in the OCOTP_CTRLn register description. Both the unlock code and address can be written in the same operation.
4. Write the data to the OCOTP_DATA register. This will automatically set OCOTP_CTRL[BUSY] and clear OCOTP_CTRL[WR_UNLOCK]. To protect programming same OTP bit twice, before a program the OCOTP will automatically read the fuse value and use it to mask program data. The controller will use masked program data to program a 32-bit word in the OTP per the address in OCOTP_CTRL[ADDR]. Field with 1's will result in that OTP bit being programmed. Field with 0's will be ignored. At the same time that the write is accepted, the controller makes an internal copy of OCOTP_CTRL[ADDR] which cannot be updated until the next write sequence is initiated. This copy guarantees that erroneous writes to OCOTP_CTRL[ADDR] will not affect an active write operation. It should also be noted that during the programming OCOTP_DATA will shift right (with zero fill). This shifting is required to program the OTP serially. During the write operation, OCOTP_DATA cannot be modified.

5. Once complete, the controller will clear OCOTP_CTRL[BUSY]. A write request to a protected or locked region will result in no OTP access and no setting of OCOTP_CTRL[BUSY]. In addition OCOTP_CTRL[ERROR] will be set. It must be cleared by software before any new write access can be issued.

It should be noted that write latencies to OTP are scaled at 10 µs per word. Write latencies will be based on the number of bits being set to 1. For example, if half the fuse bits in one word are programmed, the latency is 10 µs x 16.

For further details of OTP read/write operations see [eFUSE].

OCOTP_CTRL[ERROR] will be set under the following conditions:

- A write is performed to a shadow register during a shadow reload (essentially, while OCOTP_CTRL[RELOAD_SHADOWS] is set. In addition, the contents of the shadow register shall not be updated.
- A write is performed to a shadow register that has been locked.
- A read is performed to from a shadow register which has been read locked.
- A program is performed to a fuse word which has that locked.
- A read is performed to from a fuse word which has that read locked.

35.3.1.7 Write postamble

Due to internal electrical characteristics of the OTP during writes, all OTP operations following a write must be separated by 2 µs after the clearing of OCOTP_CTRL_BUSY following the write. This guarantees programming voltages on-chip to reach a steady state when exiting a write sequence. This includes reads, shadow reloads, or other writes. A recommended software sequence to meet the postamble requirements is as follows:

- Issue the write and poll for BUSY (as per Fuse shadow memory footprint).
- Once BUSY is clear, wait for 2 µs.
- Perform the next OTP operation.

35.3.2 OTP read/write timing parameters

Three timing fields in the OCOTP_TIMING register specify counter limit values that are used to time how long the state machine remains in the various states and also specify the STROBE signal timing. All are specified in ipg_clk cycles. Since the ipg_clk frequency is configurable, these timing parameters must be adjusted to yield the appropriate delay.

The OCOTP_TIMING[RELAX] field specifies how long to remain in states to meet setup and hold timing requirements. This parameter should be set by the following equation:

tRELAX = tHP_PG = (OCOTP_TIMING[RELAX]+1)/ipg_frequency > 16.2 ns

The OCOTP_TIMING[RELAX] field is used to calculate other setup and hold timing delays in addition to tHP_PG. For all timing to be met, this is the max delay that must be programmed.

Except for setup and hold timing delays, there are two timing parameters for the STROBE signal pulse width.

The OCOTP_TIMING[STROBE_PROG] field specifies the period of the STROBE signal during fuse writes and is given in units of ipg_clk cycles. The STROBE signal, tPGM, should be asserted high for a value between 9000 ns to 11000 ns. Even though a range is given for tPGM, it is advised to use a value of 10000 ns. Therefore, this field should be set according to the following equation:

tPGM = ((OCOTP_TIMING[STROBE_PROG]+1)- 2*(OCOTP_TIMING[RELAX]+1))/ ipg_frequency = 10000 ns

The OCOTP_TIMING[STROBE_READ] field specifies the period of the STROBE signal for fuse reads and is given in units of ipg_clk cycles. This field should be set according to the following equation:

tRD = ((OCOTP_TIMING[STROBE_READ]+1)- 2*(OCOTP_TIMING[RELAX] +1))/ipg_frequency > 36 ns.

The figure below illustrates the relationship between the STROBE signal in programming and reading mode and the timing register fields that affect it. The implementation uses one counter to generate the STROBE waveform within one period and a second counter counts the number of cycles to create for programming the designated word.

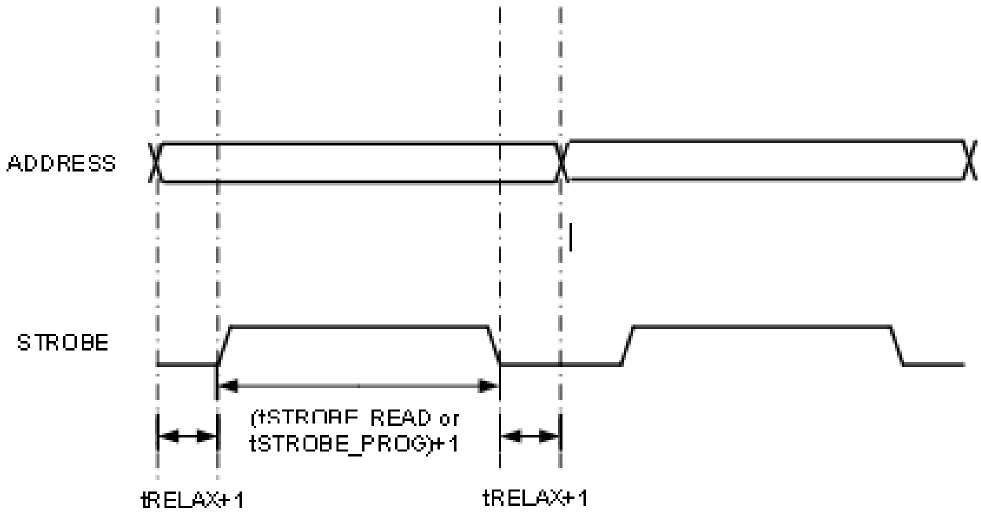


Figure 35-3. STROBE signal creation and timing

35.3.3 Behavior During Reset

The OCOTP is always active. The shadow registers automatically load the appropriate OTP contents after reset is deasserted. During this load-time OCOTP_CTRL_BUSY is set. The load time is similar to that of a "reload shadow" operation.

35.3.4 Secure JTAG control

The JTAG control fuses are used to allow or disallow JTAG access to secured resources.

Four JTAG security levels are shown in the table below.

Table 35-2. JTAG security level control bits

Security Mode	JTAG_SMODE	SEC_CONFIG	Description
No Debug	2'b11	x	The highest security level.
Secure JTAG	2'b01	x	Limit the JTAG access by using key based authentication mechanism.
JTAG Enable	2'b00	1'b1	Low Security, all JTAG features are enabled.
SCC JTAG	2'b00	1'b0	No security.

JTAG debug access may be semi-permanently enabled by blowing the jtag_bp bit (JTAG Bypass Security). This, however, can be overridden by blowing the jtag_re bit (JTAG Security Re-enable). Blowing the jtag_bp bit effectively changes the security level from 'Secure JTAG' to 'JTAG Enable', without actually modifying the JTAG_SMODE fuses.

35.4 Fuse map

See your Freescale representative for the fuse map (fuse bit definition) of the device.

35.5 OCOTP memory map/register definition

OCOTP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_5000	OTP Controller Control Register (OCOTP_CTRL)	32	R/W	0000_0000h	35.5.1/1633
400A_5004	OTP Controller Control Register (OCOTP_CTRL_SET)	32	R/W	0000_0000h	35.5.1/1633
400A_5008	OTP Controller Control Register (OCOTP_CTRL_CLR)	32	R/W	0000_0000h	35.5.1/1633
400A_500C	OTP Controller Control Register (OCOTP_CTRL_TOG)	32	R/W	0000_0000h	35.5.1/1633
400A_5010	OTP Controller Timing Register (OCOTP_TIMING)	32	R/W	0042_0078h	35.5.2/1635
400A_5020	OTP Controller Write Data Register (OCOTP_DATA)	32	R/W	0000_0000h	35.5.3/1635
400A_5030	OTP Controller Read Control Register (OCOTP_READ_CTRL)	32	R/W	0000_0000h	35.5.4/1636
400A_5040	OTP Controller Read Data Register (OCOTP_READ_FUSE_DATA)	32	R/W	0000_0000h	35.5.5/1637
400A_5060	Software Controllable Set Register (OCOTP_SCS)	32	R/W	0000_0000h	35.5.6/1637
400A_5064	Software Controllable Set Register (OCOTP_SCS_SET)	32	R/W	0000_0000h	35.5.6/1637
400A_5068	Software Controllable Set Register (OCOTP_SCS_CLR)	32	R/W	0000_0000h	35.5.6/1637
400A_506C	Software Controllable Set Register (OCOTP_SCS_TOG)	32	R/W	0000_0000h	35.5.6/1637
400A_5070	OTP Controller CRC address (OCOTP_CRC_ADDR)	32	R/W	0000_0000h	35.5.7/1638
400A_5080	OTP Controller CRC Value Register (OCOTP_CRC_VALUE)	32	R/W	0000_0000h	35.5.8/1639
400A_5090	OTP Controller Version Register (OCOTP_VERSION)	32	R	0100_0000h	35.5.9/1639
400A_5400	Value of OTP Bank0 Word0 (Lock controls) (OCOTP_LOCK)	32	R	0000_0000h	35.5.10/ 1639
400A_5410	Value of OTP Bank0 Word1 (Configuration and Manufacturing Info.) (OCOTP_CFG0)	32	R/W	0000_0000h	35.5.11/ 1642
400A_5420	Value of OTP Bank0 Word2 (Configuration and Manufacturing Info.) (OCOTP_CFG1)	32	R/W	0000_0000h	35.5.12/ 1643
400A_5450	Value of OTP Bank0 Word5 (Configuration and Manufacturing Info.) (OCOTP_CFG4)	32	R/W	0000_0000h	35.5.13/ 1643
400A_5460	Value of OTP Bank0 Word6 (Configuration and Manufacturing Info.) (OCOTP_CFG5)	32	R/W	0000_0000h	35.5.14/ 1647
400A_54F0	Value of OTP Bank1 Word7 (General Purpose Customer Defined Info.) (OCOTP_ANA2)	32	R/W	0000_0000h	35.5.15/ 1649

Table continues on the next page...

OCOTP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_5600	Value of OTP Bank4 Word0 (Secure JTAG Response Field) (OCOTP_RESP0)	32	R/W	0000_0000h	35.5.16/ 1650
400A_5610	Value of OTP Bank4 Word1 (Secure JTAG Response Field) (OCOTP_HSJC_RESP1)	32	R/W	0000_0000h	35.5.17/ 1650
400A_5620	Value of OTP Bank4 Word2 (MAC Address) (OCOTP_MAC0)	32	R/W	0000_0000h	35.5.18/ 1651
400A_5630	Value of OTP Bank4 Word3 (MAC Address) (OCOTP_MAC1)	32	R/W	0000_0000h	35.5.19/ 1651
400A_5640	Value of OTP Bank4 Word4 (MAC Address) (OCOTP_MAC2)	32	R/W	0000_0000h	35.5.20/ 1652
400A_5650	Value of OTP Bank4 Word5 (MAC Address) (OCOTP_MAC3)	32	R/W	0000_0000h	35.5.21/ 1652
400A_5660	Value of OTP Bank4 Word6 (HW Capabilities) (OCOTP_GP1)	32	R/W	0000_0000h	35.5.22/ 1653
400A_5670	Value of OTP Bank4 Word7 (HW Capabilities) (OCOTP_GP2)	32	R/W	0000_0000h	35.5.23/ 1653
400A_5880	Value of OTP Bank7 Word0 (Configuration and Manufacturing Info.) (OCOTP_TFUSE0)	32	R/W	0000_0000h	35.5.24/ 1654
400A_5890	Value of OTP Bank7 Word1 (Configuration and Manufacturing Info.) (OCOTP_TFUSE1)	32	R/W	0000_0000h	35.5.25/ 1654
400A_58B0	Value of OTP Bank7 Word3 (Configuration and Manufacturing Info.) (OCOTP_PMUR)	32	R/W	0000_0000h	35.5.26/ 1655
400A_58C0	Value of OTP Bank7 Word4 (Configuration and Manufacturing Info.) (OCOTP_PMU)	32	R/W	0000_0000h	35.5.27/ 1656
400A_58D0	Value of OTP Bank7 Word5 (Memory Related Info.) (OCOTP_RNG)	32	R/W	0000_0000h	35.5.28/ 1656
400A_58F0	Value of OTP Bank7 Word7 (Memory Related Info.) (OCOTP_VTMON)	32	R/W	0000_0000h	35.5.29/ 1657
400A_5C80	Value of OTP Bank15 Word0 (OCOTP_CRC0)	32	R/W	0000_0000h	35.5.30/ 1658
400A_5C90	Value of OTP Bank15 Word1 (OCOTP_CRC1)	32	R/W	0000_0000h	35.5.31/ 1658
400A_5CA0	Value of OTP Bank15 Word2 (OCOTP_CRC2)	32	R/W	0000_0000h	35.5.32/ 1659
400A_5CB0	Value of OTP Bank15 Word3 (OCOTP_CRC3)	32	R/W	0000_0000h	35.5.33/ 1659
400A_5CC0	Value of OTP Bank15 Word4 (OCOTP_CRC4)	32	R/W	0000_0000h	35.5.34/ 1660
400A_5CD0	Value of OTP Bank15 Word5 (OCOTP_CRC5)	32	R/W	0000_0000h	35.5.35/ 1660
400A_5CE0	Value of OTP Bank15 Word6 (OCOTP_CRC6)	32	R/W	0000_0000h	35.5.36/ 1661
400A_5CF0	Value of OTP Bank15 Word7 (OCOTP_CRC7)	32	R/W	0000_0000h	35.5.37/ 1661

35.5.1 OTP Controller Control Register (OCOTP_CTRL*n*)

The OCOTP Control and Status Register provides the necessary software interface for performing read and write operations to the On-Chip OTP. The control fields such as WR_UNLOCK, ADDR and BUSY/ERROR may be used in conjunction with the OCOTP_DATA register to perform write operations. Read operations to the OCOTP involve the ADDR and BUSY/ERROR bit fields, and the OCOTP_READ_CTRL register. The read value is saved in the OCOTP_READ_FUSE_DATA register.

Address: 400A_5000h base + 0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WR_UNLOCK															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			CRC_FAIL	CRC_TEST	RELOAD_SHADOWS	Reserved	BUSY	Reserved	ADDR						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CTRL*n* field descriptions

Field	Description
31–16 WR_UNLOCK	Write 0x3E77 to enable OTP write accesses. NOTE: This register must be unlocked on a write-by-write basis (a write is initiated when OCOTP_DATA is written), so the UNLOCK bitfield must contain the correct key value during all writes to OCOTP_DATA, otherwise a write shall not be initiated. This field is automatically cleared after a successful write completion (clearing of BUSY). 0x3E77 **KEY** — Key needed to unlock OCOTP_DATA register.
15–13 Reserved	This field is reserved. Reserved
12 CRC_FAIL	Set by controller when the calculated CRC value is not equal to appointed CRC fuse word
11 CRC_TEST	Set this bit to calculate the CRC according to the start address and the end address in the CRC_ADDR register. It compares this with the CRC fuse word selected by the CRC address in the CRC_ADDR register, and generates the value for the CRC_FAIL flag.
10 RELOAD_ SHADOWS	Set to force re-loading the shadow registers (Hardware/Software capability and LOCK). This operation will automatically set BUSY. Once the shadow registers have been re-loaded, BUSY and RELOAD_SHADOWS are automatically cleared by the controller.
9 Reserved	This field is reserved.

Table continues on the next page...

OCOTP_CTRL*n* field descriptions (continued)

Field	Description
8 BUSY	OTP controller status bit. When active, no new write access or read access to OTP (including RELOAD_SHADOWS) can be performed. It is cleared by controller when access complete. After reset (or after setting RELOAD_SHADOWS), this bit is set by the controller until the Hardware/Software and LOCK registers are successfully copied, after which time it is automatically cleared by the controller.
7 Reserved	This field is reserved. Reserved
6–0 ADDR	OTP write and read access address register. Specifies one of 128 word address locations (0x00 - 0x7f). If a valid access is accepted by the controller, the controller makes an internal copy of this value. This internal copy will not update until the access is complete.

35.5.2 OTP Controller Timing Register (OCOTP_TIMING)

This register specifies OTP Read/Write Timing Parameters

Address: 400A_5000h base + 10h offset = 400A_5010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				WAIT						STROBE_READ						RELAX				STROBE_PROG											
W																																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0

OCOTP_TIMING field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–22 WAIT	This count value specifies time interval between auto read and write access while programming the fuse. It is given in number of ipg_clk periods.
21–16 STROBE_READ	This count value specifies the strobe period in one time read OTP. Trd = ((STROBE_READ+1)- 2*(RELAX +1)) /ipg_clk_freq. It is given in number of ipg_clk periods.
15–12 RELAX	This count value specifies the time to add to all default timing parameters other than the Tpgm and Trd. It is given in number of ipg_clk periods.
11–0 STROBE_PROG	This count value specifies the strobe period in one time write OTP. Tpgm = ((STROBE_PROG+1)- 2*(RELAX+1)) /ipg_clk_freq. It is given in number of ipg_clk periods.

35.5.3 OTP Controller Write Data Register (OCOTP_DATA)

The OCOTP Data Register is used for OTP Programming.

This register is used in conjunction with OCOTP_CTRL to perform one-time writes to the OTP. Please see the "Software Write Sequence" section for operating details.

Address: 400A_5000h base + 20h offset = 400A_5020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_DATA field descriptions

Field	Description
31–0 DATA	Used to initiate a write to OTP. Please see the "Software Write Sequence" section for operating details.

35.5.4 OTP Controller Read Control Register (OCOTP_READ_CTRL)

The OCOTP Register is used for OTP Read.

This register is used in conjuction with OCOTP_CTRL to perform one time read to the OTP. Please see the "FusSoftware read Sequence" section for operating details.

The OCOTP Register is used for OTP Read

Address: 400A_5000h base + 30h offset = 400A_5030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															READ_FUSE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_READ_CTRL field descriptions

Field	Description
31–1 Reserved	This field is reserved. Reserved
0 READ_FUSE	Used to initiate a read to OTP. Please see the "Fuse and Shadow Register Read" section for operating details.

35.5.5 OTP Controller Read Data Register (OCOTP_READ_FUSE_DATA)

The data read from OTP

Address: 400A_5000h base + 40h offset = 400A_5040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_READ_FUSE_DATA field descriptions

Field	Description
31–0 DATA	The data read from OTP

35.5.6 Software Controllable Set Register (OCOTP_SCS*n*)

This register holds volatile configuration values that can be set and locked by trusted software. All values are returned to their defualt values after POR.

Address: 400A_5000h base + 60h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LOCK	SPARE														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPARE															HAB_JDE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_SCS*n* field descriptions

Field	Description
31 LOCK	When set, all of the bits in this register are locked and can not be changed through software programming. This bit is only reset after a POR is issued.
30–1 SPARE	Unallocated read/write bits for implementation specific software use.
0 HAB_JDE	HAB JTAG Debug Enable. This bit is used by the HAB to enable JTAG debugging, assuming that a properlay signed command to do so is found and validated by the HAB. The HAB must lock the register before passing control to the OS whether or not JTAG debugging has been enabled. Once JTAG is enabled by this bit, it can not be disabled unless the system is reset by POR. 0: JTAG debugging is not enabled by the HAB (it may still be enabled by other mechanisms). 1 JTAG debugging is enabled by the HAB (though this signal may be gated off)

35.5.7 OTP Controller CRC address (OCOTP_CRC_ADDR)

This register is used to calculate as part of the CRC validation. See the CRC32 test for read locking fuse area

Address: 400A_5000h base + 70h offset = 400A_5070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													CRC_ ADDR			DATA_END_ADDR								DATA_START_ADDR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC_ADDR field descriptions

Field	Description
31–19 Reserved	This field is reserved. Reserved
18–16 CRC_ADDR	Address of 32-bit CRC result for comparing
15–8 DATA_END_ ADDR	Start address of fuse location for CRC calculation
7–0 DATA_START_ ADDR	End address of fuse location for CRC calculation

35.5.8 OTP Controller CRC Value Register (OCOTP_CRC_VALUE)

The CRC32 value is based on CRC_ADDR.

Address: 400A_5000h base + 80h offset = 400A_5080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC_VALUE field descriptions

Field	Description
31–0 DATA	The CRC32 value based on CRC_ADDR

35.5.9 OTP Controller Version Register (OCOTP_VERSION)

This register always returns the OCOTP module version number.

Address: 400A_5000h base + 90h offset = 400A_5090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								STEP															
W																																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
15–0 STEP	Fixed read-only value reflecting the stepping of the RTL version.

35.5.10 Value of OTP Bank0 Word0 (Lock controls) (OCOTP_LOCK)

Copied from the OTP automatically after reset. Can be re-loaded by setting _OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 0, word 0(ADDR 0x00). This register is always read/write lock. The corresponding fuse word is always read lock.

Address: 400A_5000h base + 400h offset = 400A_5400h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		CRC_GP_HI_LOCK		CRC_GP_LO_LOCK		Reserved			MISC_CONF_LOCK	PMU_LOCK		ANALOG_LOCK		0	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	0	GP2_LOCK		GP1_LOCK		MAC_ADDR_LOCK		Reserved	SJC_RESP_LOCK	0		BOOT_CFG_LOCK		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_LOCK field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–28 CRC_GP_HI_ LOCK	Status of shadow register write and read, OTP program and read lock for upper 128 bits CRC region. When bit 1 is set, the reading and writing of this region's OTP fuse and reading of shadow register are blocked.When bit 0 is set, the writing of this region's shadow register and OTP fuse are blocked.
27–26 CRC_GP_LO_ LOCK	Status of shadow register write and read, OTP program and read lock for lower 128 bits CRC region. When bit 1 is set, the reading and writing of this region's OTP fuse and reading of shadow register are blocked.When bit 0 is set, the writing of this region's shadow register and OTP fuse are blocked.
25–23 Reserved	This field is reserved. Reserved
22 MISC_CONF_ LOCK	Status of shadow register and OTP write lock for misc_conf region. When set, the writing of this region's shadow register and OTP fuse word are blocked.
21–20 PMU_LOCK	Status of shadow register and OTP write lock for PMU region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
19–18 ANALOG_LOCK	Status of shadow register and OTP write lock for analog region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.

Table continues on the next page...

OCOTP_LOCK field descriptions (continued)

Field	Description
17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16–15 Reserved	This field is reserved. Reserved
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–12 GP2_LOCK	Lock for General Purpose fuse register #2 (GP2) Status of shadow register and OTP write lock for GP2 region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
11–10 GP1_LOCK	Lock for General Purpose fuse register #1 (GP1) Status of shadow register and OTP write lock for GP1 region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
9–8 MAC_ADDR_ LOCK	Lock MAC_ADDR fuses. Status of shadow register and OTP write lock for mac_addr region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
7 Reserved	Reserved. This field is reserved. Reading this bit will get a 1.
6 SJC_RESP_ LOCK	Status of shadow register read and write, OTP read and write lock for sjc_resp region. When set, the writing of this region's shadow register and OTP fuse word are blocked. The read of this region's shadow register and OTP fuse word are also blocked.
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–2 BOOT_CFG_ LOCK	Perform lock on BOOT related fuses. Status of shadow register and OTP write lock for boot_cfg region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
1–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

35.5.11 Value of OTP Bank0 Word1 (Configuration and Manufacturing Info.) (OCOTP_CFG0)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadows the OTP fuse BANK 0, word 1 (ADDR 0x01). This register is read-only after TESTER_LOCK[1] is set. The corresponding fuse word is also read-only after TESTER_LOCK[0] is set.

Address: 400A_5000h base + 410h offset = 400A_5410h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SJC_CHALL																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG0 field descriptions

Field	Description
31–0 SJC_CHALL	FSL-wide unique, encoded LOT ID STD II/SJC CHALLENGE/ Unique ID

35.5.12 Value of OTP Bank0 Word2 (Configuration and Manufacturing Info.) (OCOTP_CFG1)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadows the OTP fuse BANK 0, word 2 (ADDR 0x02). This register is read-only after TESTER_LOCK[1] is set. The corresponding fuse word is also read-only after TESTER_LOCK[0] is set.

Address: 400A_5000h base + 420h offset = 400A_5420h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SJC_CHALL																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG1 field descriptions

Field	Description
31–0 SJC_CHALL	The wafer number of the wafer on which the device was fabricated/SJC CHALLENGE/ Unique ID

35.5.13 Value of OTP Bank0 Word5 (Configuration and Manufacturing Info.) (OCOTP_CFG4)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 0, word 5 (ADDR 0x05). This register is read-only after BOOT_CFG_LOCK [1] is set. The corresponding fuse word is also read-only after BOOT_CFG_LOCK [0] is set.

Address: 400A_5000h base + 450h offset = 400A_5450h

Bit	31–24	23–16	15–8	7–0
R/W	BOOT_CFG4	BOOT_CFG3	BOOT_CFG2	BOOT_CFG1
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

OCOTP_CFG4 field descriptions

Field	Description
31–24 BOOT_CFG4	See table below
23–16 BOOT_CFG3	See table below
15–8 BOOT_CFG2	See table below

31–24 BOOT_CFG4

	BOOT_CFG4[7]	BOOT_CFG4[6]	BOOT_CFG4[5]	BOOT_CFG4[4]	BOOT_CFG4[3]	BOOT_CFG4[2]	BOOT_CFG4[1]	BOOT_CFG4[0]
	Infinite Loop (Debug USE only) 0 - Disable 1- Enable	EEPROM Recovery Enable '0' - Disabled '1' - Enabled"	CS select (SPI only): 00 - CS#0 (default) 01 - CS#1 10 - CS#2 11 - CS#3		SPI Addressing: 0 - 2-bytes (16-bit) 1 - 3-bytes (24-bit)	Port Select: 000 - SPI0 001 - SPI1 010 - SPI2 011 - SPI3 100 - I2C0 101 - I2C1 110 - I2C2 111 - I2C3		

23–16 BOOT_CFG3

	BOOT_CFG3[7]	BOOT_CFG3[6]	BOOT_CFG3[5]	BOOT_CFG3[4]	BOOT_CFG3[3]	BOOT_CFG3[2]	BOOT_CFG3[1]	BOOT_CFG3[0]
	Reserved	BT_MMU_DISABLE	UART Select : 00 : UART0 01 : UART1 10: UART2 11: UART 3		Reserved	DDR Clock Mode 0 = Asynchronous 1 = Synchronous	Reserved	Reserved

15–8 BOOT_CFG2

	BOOT_CFG2[7]	BOOT_CFG2[6]	BOOT_CFG2[5]	BOOT_CFG2[4]	BOOT_CFG2[3]	BOOT_CFG2[2]	BOOT_CFG2[1]	BOOT_CFG2[0]
QuadSPI	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved		

Table continues on the next page...

OCOTP_CFG4 field descriptions

Field	Description								
		BOOT_CFG2[7]	BOOT_CFG2[6]	BOOT_CFG2[5]	BOOT_CFG2[4]	BOOT_CFG2[3]	BOOT_CFG2[2]	BOOT_CFG2[1]	BOOT_CFG2[0]
	NOR Flash	Reserved	Reserved	FB_ASET 00 Assert FB_CSn on the first rising clock edge after the address is asserted. 01 Assert FB_CSn on the second rising clock edge after the address is asserted. 10 Assert FB_CSn on the third rising clock edge after the address is asserted. 11 Assert FB_CSn on the fourth rising clock edge after the address is asserted.	FB_ASET 00 Assert FB_CSn on the first rising clock edge after the address is asserted. 01 Assert FB_CSn on the second rising clock edge after the address is asserted. 10 Assert FB_CSn on the third rising clock edge after the address is asserted. 11 Assert FB_CSn on the fourth rising clock edge after the address is asserted.	FB_ELE 0 :FB_TS/ALE assert for one cycle 1: FB_TS/ALE asserted until the first positive edge after FB_CS0 asserts	Reserved	FB_BEM 0 FB_BE is asserted for data write only. 1 FB_BE is asserted for data read and write accesses.	FB_TA 0 Internal Transfer ACK 1 External Transfer ACK
	Serial-ROM (SPI/IIC)	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	FlexCAN	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Table continues on the next page...

OCOTP_CFG4 field descriptions (continued)

Field	Description
	(see table below)

	BOOT_CFG2[7]	BOOT_CFG2[6]	BOOT_CFG2[5]	BOOT_CFG2[4]	BOOT_CFG2[3]	BOOT_CFG2[2]	BOOT_CFG2[1]	BOOT_CFG2[0]
SD/eSD	Reserved		Bus Width: 0 - 1-bit 1 - 4-bit	Reserved	Port Select: 0 - eSDHC0 1 - eSDHC1	Reserved	Reserved	Override Pad Settings (using PAD_SETTINGS value)
MMC/ eMMC	Reserved	Bus Width: 00 - 1-bit 01 - 4-bit 10 - 8-bit 11 - Reserved for Future.		Reserved	Port Select: 0 - eSDHC0 1 - eSDHC1	Reserved	Fast Boot Acknowledge Disable: 0 - Boot Ack Enabled 1 - Boot Ack Disabled	Override Pad Settings (using PAD_SETTINGS value)
NAND	Reserved			BOOT_SEARCH_COUNT: 00 - 2 01 - 2 10 - 4 11 - 8		Pages In Block: 00 - 128 01 - 64 10 - 32 11 - Reserved		Override Pad Settings (using PAD_SETTINGS value)

7–0 BOOT_CFG1

	BOOT_CFG1[7]	BOOT_CFG1[6]	BOOT_CFG1[5]	BOOT_CFG1[4]	BOOT_CFG1[3]	BOOT_CFG1[2]	BOOT_CFG1[1]	BOOT_CFG1[0]
QuadSPI	0	0	0	0	Reserved	Reserved	QXIP Instance 0: QuadSPI0 1: QuadSPI1	Reserved
NOR Flash	0	0	0	1	"FBCS 0 = boot from FB_CS0 1 = boot from FB_CS1"	"BOOTPS 1x = 16-bit 01 = 8-bit 00 = 32 bit"		"FMNM 0 : Non Muxed Mode 1: Muxed Mode"
Serial-ROM(SPI/ IIC)	0	0	1	0	Reserved	Reserved	Reserved	Reserved

Table continues on the next page...

OCOTP_CFG4 field descriptions (continued)

	BOOT_CFG1[7]	BOOT_CFG1[6]	BOOT_CFG1[5]	BOOT_CFG1[4]	BOOT_CFG1[3]	BOOT_CFG1[2]	BOOT_CFG1[1]	BOOT_CFG1[0]
FlexCAN	0	0	1	1	Reserved	Reserved	Reserved	"FCPS: 0: Boot from Flexcan 0 1: Boot from Flexcan 1"
Reserved	0	1	0	0	Reserved	Reserved	Reserved	Reserved
Reserved	0	1	0	1	Reserved	Reserved	Reserved	Reserved
SD/eSD	0	1	1	0	Fast BOOT 0 - Normal Mode 1 - Fast Mode	Reserved	SDSpeed 1 - Normal 0 - High	Reserved
MMC/ eMMC	0	1	1	1	Fast BOOT 0 - Normal Mode 1 - Fast Mode	Reserved	SD/MMC Speed 0 - High 1- Normal	Reserved
NAND	1	Freq Select 0 = Normal Mode (33Mhz) 1 = Fast Mode(40 Mhz)	Fast BOOT Ack 0 = Slow 1 = Fast Boot Ack	Reserved	Nand Chip Enable 0 : NFC_CE0_b 1: NFC_CE1_b	Nand Data Width 0 : 8 bits 1 : 16 bits	Nand_Row_address_bytes: 00 - 3 01 - 2 10 - Reserved 11 - Reserved	

35.5.14 Value of OTP Bank0 Word6 (Configuration and Manufacturing Info.) (OCOTP_CFG5)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 0, word 6 (ADDR 0x06). This register is read-only after BOOT_CFG_LOCK [1] is set. The corresponding fuse word is also read-only after BOOT_CFG_LOCK [0] is set.

Address: 400A_5000h base + 460h offset = 400A_5460h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				JTAG_HEO	KTE	0		JTAG_SMODE		WDOG_ENABLE	SJC_DISABLE	0	Reserved	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											BT_FUSE_SEL	DIR_BT_DIS	Reserved	SEC_CONFIG	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CFG5 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 JTAG_HEO	JTAG HAB Enable Override. Disallows HAB JTAG enabling. The HAB may normally enable JTAG debugging by means of the HAB_JDE-bit in the OCOTP SCS register. The JTAG_HEO-bit can override this behavior. 0 HAB may enable JTAG debug access 1 HAB JTAG enable is overridden (HAB may not enable JTAG debug access)
26 KTE	Kill Trace Enable. Enables tracing capability on ETM, and other modules. 0 Bus tracing is allowed 1 Bus tracing is allowed in case security state as defined by Secure JTAG allows it (for example, JTAG_ENABLE or NO_DEBUG)
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–22 JTAG_SMODE	JTAG Security Mode. Controls the security mode of the JTAG debug interface. 00 JTAG enable mode 01 Secure JTAG mode 11 No debug mode
21 WDOG_ENABLE	Watchdog Enable. Used to specify whether to enable / not watchdog at boot. 0 Watch-Dog is disabled. 1 Watch-Dog is enabled.

Table continues on the next page...

OCOTP_CFG5 field descriptions (continued)

Field	Description
20 SJC_DISABLE	Secure JTAG Controller module. This fuse is used to create highest JTAG security level, where JTAG is totally blocked. 0 Secure JTAG Controller is enabled 1 Secure JTAG Controller is disabled
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 Reserved	This field is reserved.
17–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 BT_FUSE_SEL	Determines, whether using fuses for boot configuration, or GPIO /Serial loader. 0=Boot mode configuration is taken from GPIOs. 1=Boot mode configuration is taken from fuses. 0 - Boot using Serial Loader (USB) 1- Boot mode configuration is taken from fuses. If boot_mode="00" (Development) If boot_mode="10" (Production)
3 DIR_BT_DIS	Direct Boot Disable. 0 Enable 1 Disable
2 Reserved	This field is reserved.
1 SEC_CONFIG	Security Configuration (with SEC_CONFIG[0]). 00 - FAB (Open) 01 - Open - allows any code to be flashed and executed, even if it has no valid signature. 1x - Closed (Security On)
0 Reserved	This field is reserved.

35.5.15 Value of OTP Bank1 Word7 (General Purpose Customer Defined Info.) (OCOTP_ANA2)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 1, word 7 (ADDR 0x0F). This register is read-only after ANALOG_LOCK[1] is set. The corresponding fuse word is also read-only after ANALOG_LOCK[0] is set.

Address: 400A_5000h base + 4F0h offset = 400A_54F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	USB0_PID																USB_VID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_ANA2 field descriptions

Field	Description
31–16 USB0_PID	USB Product ID.
15–0 USB_VID	USB Vendor ID.

35.5.16 Value of OTP Bank4 Word0 (Secure JTAG Response Field) (OCOTP_RESP0)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 4, word 0 (ADDR 0x20). This register and corresponding fuse is neither readable nor writable after SJC_RESP_LOCK bit is set.

Address: 400A_5000h base + 600h offset = 400A_5600h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_RESP0 field descriptions

Field	Description
31–0 BITS	Shadow register for the SJC_RESP Key word0 (OTP Bank 4, word 0 (ADDR = 0x20)). These bits can be not read and written after the SJC_RESP_LOCK bit is set. If read, returns 0xBADA_BADA and sets OCOTP_CTRL[ERROR].

35.5.17 Value of OTP Bank4 Word1 (Secure JTAG Response Field) (OCOTP_HSJC_RESP1)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 4, word 1 (ADDR 0x21). This register and corresponding fuse is neither readable nor writable after SJC_RESP_LOCK bit is set.

Address: 400A_5000h base + 610h offset = 400A_5610h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HSJC_RESP1 field descriptions

Field	Description
31–0 BITS	Shadow register for the SJC_RESP Key word1 (OTP Bank 4, word 1 (ADDR = 0x21)). These bits can be not read and written after the SJC_RESP_LOCK bit is set. If read, returns 0xBADA_BADA and sets OCOTP_CTRL[ERROR].

35.5.18 Value of OTP Bank4 Word2 (MAC Address) (OCOTP_MAC0)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 4, word 2 (ADDR 0x22). This register is read-only after MAC_ADDR_LOCK [1] is set. The corresponding fuse word is also read-only after MAC_ADDR_LOCK[0] is set.

Address: 400A_5000h base + 620h offset = 400A_5620h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MAC0 field descriptions

Field	Description
31–0 BITS	Reflects value of OTP Bank 4, word 2 (ADDR = 0x22).

35.5.19 Value of OTP Bank4 Word3 (MAC Address) (OCOTP_MAC1)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 4, word 3 (ADDR 0x23). This register is read-only after MAC_ADDR_LOCK[1] is set. The corresponding fuse word is also read-only after MAC_ADDR_LOCK[0] is set.

Address: 400A_5000h base + 630h offset = 400A_5630h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MAC1 field descriptions

Field	Description
31–0 BITS	Reflects value of OTP Bank 4, word 3 (ADDR = 0x23).

35.5.20 Value of OTP Bank4 Word4 (MAC Address) (OCOTP_MAC2)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 4, word 4 (ADDR 0x24). This register is read-only after MAC_ADDR_LOCK[1] is set. The corresponding fuse word is also read-only after MAC_ADDR_LOCK[0] is set.

Address: 400A_5000h base + 640h offset = 400A_5640h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MAC2 field descriptions

Field	Description
31–0 BITS	Reflects value of OTP Bank 4, word 4 (ADDR = 0x24).

35.5.21 Value of OTP Bank4 Word5 (MAC Address) (OCOTP_MAC3)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 4, word 5 (ADDR 0x25). This register is read-only after MAC_ADDR_LOCK[1] is set. The corresponding fuse word is also read-only after MAC_ADDR_LOCK[0] is set.

Address: 400A_5000h base + 650h offset = 400A_5650h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_MAC3 field descriptions

Field	Description
31–0 BITS	Reflects value of OTP Bank 4, word 5 (ADDR = 0x25).

35.5.22 Value of OTP Bank4 Word6 (HW Capabilities) (OCOTP_GP1)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 4, word 6 (ADDR 0x26). This register is read-only after GP1_LOCK[1] is set. The corresponding fuse word is also read-only after GP1_LOCK[0] is set.

Address: 400A_5000h base + 660h offset = 400A_5660h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_GP1 field descriptions

Field	Description
31–0 BITS	Reflects value of OTP Bank 4, word 6 (ADDR = 0x26).

35.5.23 Value of OTP Bank4 Word7 (HW Capabilities) (OCOTP_GP2)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 4, word 7 (ADDR 0x27). This register is read-only after GP2_LOCK[1] is set. The corresponding fuse word is also read-only after GP2_LOCK[0] is set.

Address: 400A_5000h base + 670h offset = 400A_5670h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_GP2 field descriptions

Field	Description
31–0 BITS	Reflects value of OTP Bank 4, word 7 (ADDR = 0x27).

35.5.24 Value of OTP Bank7 Word0 (Configuration and Manufacturing Info.) (OCOTP_TFUSE0)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 7, word 0 (ADDR 0x38). This register is read-only after TESTER_LOCK[1] is set. The corresponding fuse word is also read-only after TESTER_LOCK[0] is set.

Address: 400A_5000h base + 880h offset = 400A_5880h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								Reserved							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_TFUSE0 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 Reserved	This field is reserved. Reserved.

35.5.25 Value of OTP Bank7 Word1 (Configuration and Manufacturing Info.) (OCOTP_TFUSE1)

Copied from the OTP automatically after reset. Can be re-loaded by setting HW_OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 7, word 1 (ADDR 0x39). This register is read-only after TESTER_LOCK[1] is set. The corresponding fuse word is also read-only after TESTER_LOCK[0] is set.

Address: 400A_5000h base + 890h offset = 400A_5890h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								Reserved							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_TFUSE1 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 Reserved	This field is reserved. Reserved.

35.5.26 Value of OTP Bank7 Word3 (Configuration and Manufacturing Info.) (OCOTP_PMUR)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 7, word 3 (ADDR 0x3B). This register is read-only after PMU_LOCK[1] is set. The corresponding fuse word is also read-only after PMU_LOCK[0] is set.

Address: 400A_5000h base + 8B0h offset = 400A_58B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				Reserved				0								Reserved												0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_PMUR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 Reserved	This field is reserved. Reserved.
23–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–4 Reserved	This field is reserved. Reserved.
3–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

35.5.27 Value of OTP Bank7 Word4 (Configuration and Manufacturing Info.) (OCOTP_PMU)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 7, word 4 (ADDR 0x3C). This register is read-only after PMU_LOCK[1] is set. The corresponding fuse word is also read-only after PMU_LOCK[0] is set.

Address: 400A_5000h base + 8C0h offset = 400A_58C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												0	Reserved		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_PMU field descriptions

Field	Description
31–4 Reserved	This field is reserved. Reserved.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 Reserved	This field is reserved. Reserved.

35.5.28 Value of OTP Bank7 Word5 (Memory Related Info.) (OCOTP_RNG)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 7, word 5 (ADDR 0x3D). This register is read-only after ANALOG_LOCK[1] is set. The corresponding fuse word is also read-only after ANALOG_LOCK[0] is set.

Address: 400A_5000h base + 8D0h offset = 400A_58D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RNG_TRIM								Reserved								ADC1_CAL				ADC0_CAL				0				VADC_ BANDGAP			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_RNG field descriptions

Field	Description
31–24 RNG_TRIM	Trims bits for RNG in CAAM. 6-bits of the field control the entropy delay value, freq_count_max and freq_count_min: 2-bits control the oscillator output divider.
23–16 Reserved	This field is reserved.
15–12 ADC1_CAL	ADC1 Calibration value
11–8 ADC0_CAL	Calibration 0 for ADC0
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 VADC_ BANDGAP	Trim values for VADC bandgap.

35.5.29 Value of OTP Bank7 Word7 (Memory Related Info.) (OCOTP_VTMON)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 7, word 7 (ADDR 0x3F). This register is read-only after ANALOG_LOCK[1] is set. The corresponding fuse word is also read-only after ANALOG_LOCK[0] is set.

Address: 400A_5000h base + 8F0h offset = 400A_58F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	USB1_PID																0										VOLT_TEMP_ TAMPER_PROG					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_VTMON field descriptions

Field	Description
31–16 USB1_PID	USB1 PID
15–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

OCOTP_VTMON field descriptions (continued)

Field	Description
5–0 VOLT_TEMP_ TAMPER_PROG	Voltage and Temperature Tamper Detect PROG trims.

35.5.30 Value of OTP Bank15 Word0 (OCOTP_CRC0)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 15, word 0 (ADDR 0x78). The reading of shadow registers and reading and writing of corresponding fuse are blocked after CRC_GP_LO_LOCK[1] is set. When CRC_GP_LO_LOCK[0] is set, the writing of this shadow register and corresponding fuse are blocked.

Address: 400A_5000h base + C80h offset = 400A_5C80h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC0 field descriptions

Field	Description
31–0 CRC32	Reflects the calculated CRC32 for the pre-defined fuse area. Refer to CRC32 test for read locking fuse area for more details.

35.5.31 Value of OTP Bank15 Word1 (OCOTP_CRC1)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 15, word 1 (ADDR 0x79). The reading of shadow registers and reading and writing of corresponding fuse are blocked after CRC_GP_LO_LOCK[1] is set. When CRC_GP_LO_LOCK[0] is set, the writing of this shadow register and corresponding fuse are blocked.

Address: 400A_5000h base + C90h offset = 400A_5C90h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC1 field descriptions

Field	Description
31–0 CRC32	Reflects the calculated CRC32 for the pre-defined fuse area. Refer to CRC32 test for read locking fuse area for more details.

35.5.32 Value of OTP Bank15 Word2 (OCOTP_CRC2)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 15, word 2 (ADDR 0x7A). The reading of shadow registers and reading and writing of corresponding fuse are blocked after CRC_GP_LO_LOCK[1] is set. When CRC_GP_LO_LOCK[0] is set, the writing of this shadow register and corresponding fuse are blocked.

Address: 400A_5000h base + CA0h offset = 400A_5CA0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC2 field descriptions

Field	Description
31–0 CRC32	Reflects the calculated CRC32 for the pre-defined fuse area. Refer to CRC32 test for read locking fuse area for more details.

35.5.33 Value of OTP Bank15 Word3 (OCOTP_CRC3)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 15, word 3 (ADDR 0x7B). The reading of shadow registers and reading and writing of corresponding fuse are blocked after CRC_GP_LO_LOCK[1] is set. When CRC_GP_LO_LOCK[0] is set, the writing of this shadow register and corresponding fuse are blocked.

Address: 400A_5000h base + CB0h offset = 400A_5CB0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC3 field descriptions

Field	Description
31–0 CRC32	Reflects the calculated CRC32 for the pre-defined fuse area. Refer to CRC32 test for read locking fuse area for more details.

35.5.34 Value of OTP Bank15 Word4 (OCOTP_CRC4)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 15, word 4 (ADDR 0x7C). The reading of shadow registers and reading and writing of corresponding fuse are blocked after CRC_GP_HI_LOCK[1] is set. When CRC_GP_HI_LOCK[0] is set, the writing of this shadow register and corresponding fuse are blocked.

Address: 400A_5000h base + CC0h offset = 400A_5CC0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC4 field descriptions

Field	Description
31–0 CRC32	Reflects the calculated CRC32 for the pre-defined fuse area. Refer to CRC32 test for read locking fuse area for more details.

35.5.35 Value of OTP Bank15 Word5 (OCOTP_CRC5)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 15, word 5 (ADDR 0x7D). The reading of shadow registers and reading and writing of corresponding fuse are blocked after CRC_GP_HI_LOCK[1] is set. When CRC_GP_HI_LOCK[0] is set, the writing of this shadow register and corresponding fuse are blocked.

Address: 400A_5000h base + CD0h offset = 400A_5CD0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC5 field descriptions

Field	Description
31–0 CRC32	Reflects the calculated CRC32 for the pre-defined fuse area. Refer to CRC32 test for read locking fuse area for more details.

35.5.36 Value of OTP Bank15 Word6 (OCOTP_CRC6)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 15, word 6 (ADDR 0x7E). The reading of shadow registers and reading and writing of corresponding fuse are blocked after CRC_GP_HI_LOCK[1] is set. When CRC_GP_HI_LOCK[0] is set, the writing of this shadow register and corresponding fuse are blocked.

Address: 400A_5000h base + CE0h offset = 400A_5CE0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC6 field descriptions

Field	Description
31–0 CRC32	Reflects the calculated CRC32 for the pre-defined fuse area. Refer to CRC32 test for read locking fuse area for more details.

35.5.37 Value of OTP Bank15 Word7 (OCOTP_CRC7)

Copied from the OTP automatically after reset. Can be re-loaded by setting OCOTP_CTRL[RELOAD_SHADOWS]

This register shadowed the OTP fuse BANK 15, word 7 (ADDR 0x7F). The reading of shadow registers and reading and writing of corresponding fuse are blocked after CRC_GP_HI_LOCK[1] is set. When CRC_GP_HI_LOCK[0] is set, the writing of this shadow register and corresponding fuse are blocked.

Address: 400A_5000h base + CF0h offset = 400A_5CF0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC32																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_CRC7 field descriptions

Field	Description
31–0 CRC32	Reflects the calculated CRC32 for the pre-defined fuse area. Refer to CRC32 test for read locking fuse area for more details.

Chapter 36
12-bit Digital-to-Analog Converter (DAC)

36.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The 12-bit digital-to-analog converter (DAC) is a low-power, general-purpose DAC. The output of the DAC can be placed on an external pin or set as one of the inputs to the analog comparator, op-amps, or ADC.

36.2 Features

The features of the DAC module include:

- On-chip programmable reference generator output. The voltage output range is from 1⁄4096 V_{in} to V_{in}, and the step is 1⁄4096 V_{in}, where V_{in} is the input voltage.
- V_{in} can be selected from two reference sources
- Static operation in Normal Stop mode
- 16-word data buffer supported with configurable watermark and multiple operation modes
- DMA support

36.3 Block diagram

The block diagram of the DAC module is as follows:

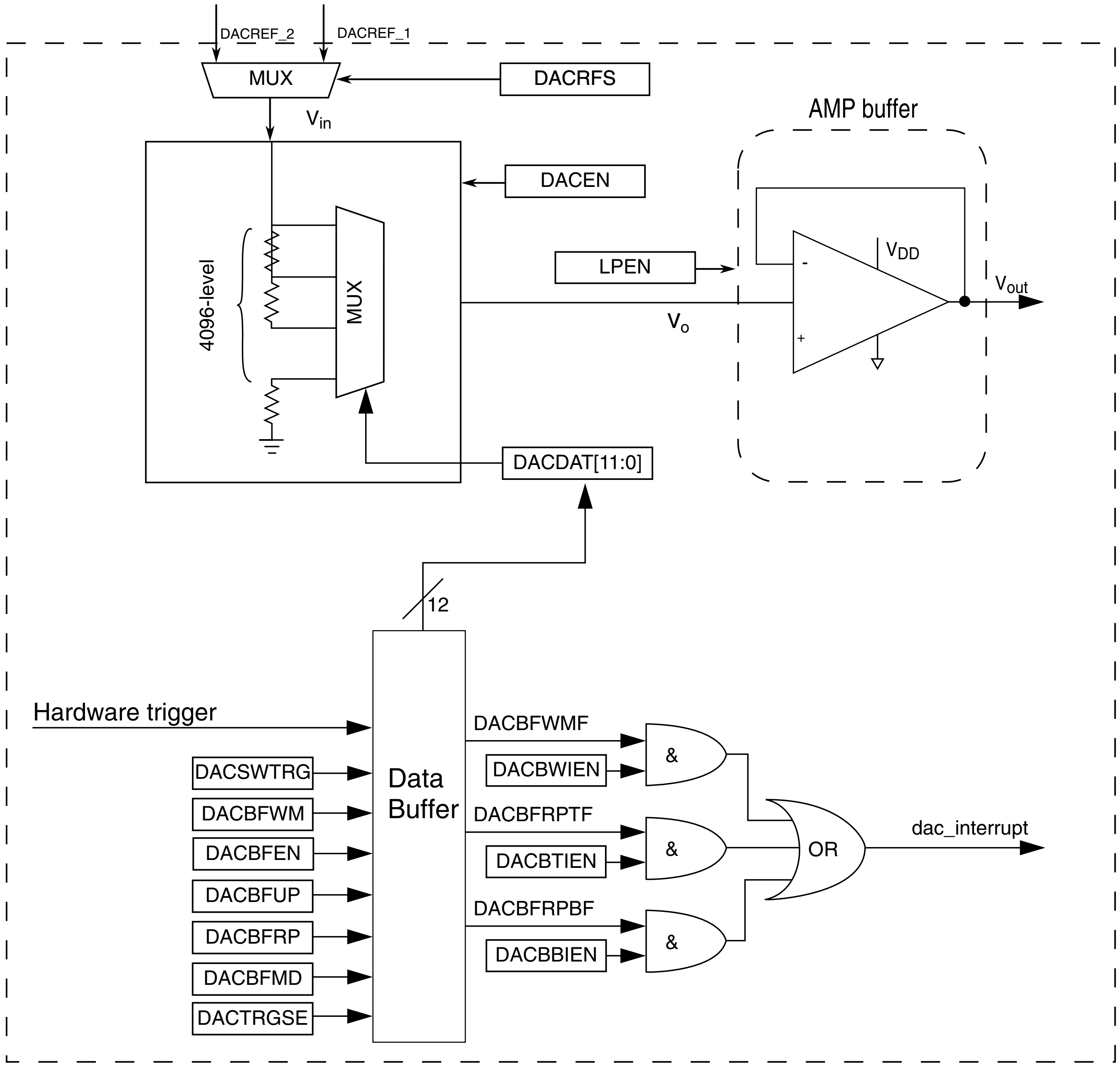


Figure 36-1. DAC block diagram

36.4 Memory map/register definition

The DAC has registers to control analog comparator and programmable voltage divider to perform the digital-to-analog functions.

DAC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_C000	DAC Data Register (DAC0_DAT0)	32	R/W	0000_0000h	36.4.1/1665
400C_C004	DAC Data Register (DAC0_DAT1)	32	R/W	0000_0000h	36.4.1/1665
400C_C008	DAC Data Register (DAC0_DAT2)	32	R/W	0000_0000h	36.4.1/1665
400C_C00C	DAC Data Register (DAC0_DAT3)	32	R/W	0000_0000h	36.4.1/1665
400C_C010	DAC Data Register (DAC0_DAT4)	32	R/W	0000_0000h	36.4.1/1665
400C_C014	DAC Data Register (DAC0_DAT5)	32	R/W	0000_0000h	36.4.1/1665
400C_C018	DAC Data Register (DAC0_DAT6)	32	R/W	0000_0000h	36.4.1/1665
400C_C01C	DAC Data Register (DAC0_DAT7)	32	R/W	0000_0000h	36.4.1/1665
400C_C020	DAC Status and Control Register (DAC0_STATCTRL)	32	R/W	See section	36.4.2/1666
400C_D000	DAC Data Register (DAC1_DAT0)	32	R/W	0000_0000h	36.4.1/1665
400C_D004	DAC Data Register (DAC1_DAT1)	32	R/W	0000_0000h	36.4.1/1665
400C_D008	DAC Data Register (DAC1_DAT2)	32	R/W	0000_0000h	36.4.1/1665
400C_D00C	DAC Data Register (DAC1_DAT3)	32	R/W	0000_0000h	36.4.1/1665
400C_D010	DAC Data Register (DAC1_DAT4)	32	R/W	0000_0000h	36.4.1/1665
400C_D014	DAC Data Register (DAC1_DAT5)	32	R/W	0000_0000h	36.4.1/1665
400C_D018	DAC Data Register (DAC1_DAT6)	32	R/W	0000_0000h	36.4.1/1665
400C_D01C	DAC Data Register (DAC1_DAT7)	32	R/W	0000_0000h	36.4.1/1665
400C_D020	DAC Status and Control Register (DAC1_STATCTRL)	32	R/W	See section	36.4.2/1666

36.4.1 DAC Data Register (DACx_DAT*n*)

Address: Base address + 0h offset + (4d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DATA1												0				DATA0											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DACx_DAT*n* field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 DATA1	When the DAC buffer is not enabled, this field controls the output voltage based on the following formula: $V_{out} = V_{in} * (1 + DAC\_DATn[DATA1])/4096$ When the DAC buffer is enabled, this field is mapped to the 16-word buffer.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 DATA0	When the DAC buffer is not enabled, this field controls the output voltage based on the following formula: $V_{out} = V_{in} * (1 + DAC\_DATn[DATA0])/4096$ When the DAC buffer is enabled, this field is mapped to the 16-word buffer.

36.4.2 DAC Status and Control Register (DACx_STATCTRL)

If DMA is enabled, the flags can be cleared automatically by DMA when the DMA request is done. Writing 0 to a field clears it whereas writing 1 has no effect. After reset, DACBFRPTF is set and can be cleared by software, if needed. The flags are set only when the data buffer status is changed.

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DACBFRP				DACBFUP				DMAEN	0		DACBFWM		DACBFMD		DACBFEN
W																
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DACEN	DACRFS	DACTRGSEL	0	LPEN	DACBWIEN	DACBTIEN	DACBBIEN	0					DACBFWMF	DACBFRPTF	DACBFRPBF
W				DACSWTRG												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

DACx_STATCTRL field descriptions

Field	Description
31–28 DACBFRP	DAC Buffer Read Pointer Keeps the current value of the buffer read pointer.
27–24 DACBFUP	DAC Buffer Upper Limit Selects the upper limit of the DAC buffer. The buffer read pointer cannot exceed it.
23 DMAEN	DMA Enable Select 0 DMA is disabled. 1 DMA is enabled. When DMA is enabled, the DMA request will be generated by original interrupts. The interrupts will not be presented on this module at the same time.

Table continues on the next page...

DACx_STATCTRL field descriptions (continued)

Field	Description
22–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–19 DACBFWM	DAC Buffer Watermark Select Controls when DACBFWMF will be set. When the DAC buffer read pointer reaches the word defined by this field, which is 1–4 words away from the upper limit (DACBUP), DACBFWMF will be set. This allows user configuration of the watermark interrupt. 00 1 word 01 2 words 10 3 words 11 4 words
18–17 DACBFMD	DAC Buffer Work Mode Select 00 Normal mode 01 Swing mode 10 One-Time Scan mode 11 Reserved
16 DACBFEN	DAC Buffer Enable 0 Buffer read pointer is disabled. The converted data is always the first word of the buffer. 1 Buffer read pointer is enabled. The converted data is the word that the read pointer points to. It means converted data can be from any word of the buffer.
15 DACEN	DAC Enable Starts the Programmable Reference Generator operation. 0 The DAC system is disabled. 1 The DAC system is enabled.
14 DACRFS	DAC Reference Select 0 The DAC selects DACREF_1 as the reference voltage. 1 The DAC selects DACREF_2 as the reference voltage.
13 DACTRGSEL	DAC Trigger Select 0 The DAC hardware trigger is selected. 1 The DAC software trigger is selected.
12 DACSWTRG	DAC Software Trigger Active high. This is a write-only field, which always reads 0. If DAC software trigger is selected and buffer is enabled, writing 1 to this field will advance the buffer read pointer once. 0 The DAC soft trigger is not valid. 1 The DAC soft trigger is valid.
11 LPEN	DAC Low Power Control **NOTE:** See the 12-bit DAC electrical characteristics of the device data sheet for details on the impact of the modes below.

Table continues on the next page...

DACx_STATCTRL field descriptions (continued)

Field	Description
	0 High-Power mode 1 Low-Power mode
10 DACBWIEN	DAC Buffer Watermark Interrupt Enable 0 The DAC buffer watermark interrupt is disabled. 1 The DAC buffer watermark interrupt is enabled.
9 DACBTIEN	DAC Buffer Read Pointer Top Flag Interrupt Enable 0 The DAC buffer read pointer top flag interrupt is disabled. 1 The DAC buffer read pointer top flag interrupt is enabled.
8 DACBBIEN	DAC Buffer Read Pointer Bottom Flag Interrupt Enable 0 The DAC buffer read pointer bottom flag interrupt is disabled. 1 The DAC buffer read pointer bottom flag interrupt is enabled.
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 DACBFWMF	DAC Buffer Watermark Flag 0 The DAC buffer read pointer has not reached the watermark level. 1 The DAC buffer read pointer has reached the watermark level.
1 DACBFRPTF	DAC Buffer Read Pointer Top Position Flag 0 The DAC buffer read pointer is not zero. 1 The DAC buffer read pointer is zero.
0 DACBFRPBF	DAC Buffer Read Pointer Bottom Position Flag 0 The DAC buffer read pointer is not equal to DACBFUP. 1 The DAC buffer read pointer is equal to DACBFUP.

36.5 Functional description

The 12-bit DAC module can select one of the two reference inputs—DACREF_1 and DACREF_2 as the DAC reference voltage, V_{in} by STATCTRL[DACRFS]. See the module introduction for information on the source for DACREF_1 and DACREF_2.

When the DAC is enabled, it converts the data in DACDAT0[11:0] or the data from the DAC data buffer to a stepped analog output voltage. The output voltage range is from V_{in} to $V_{in}/4096$, and the step is $V_{in}/4096$.

36.5.1 DAC data buffer operation

When the DAC is enabled and the buffer is not enabled, the DAC module always converts the data in DAT0 to analog output voltage.

When both the DAC and the buffer are enabled, the DAC converts the data in the data buffer to analog output voltage. The data buffer read pointer advances to the next word whenever any hardware or software trigger event occurs. Refer to Introduction for the hardware trigger connection.

The data buffer can be configured to operate in Normal mode, Swing mode, or One-Time Scan mode. When the buffer operation is switched from one mode to another, the read pointer does not change. The read pointer can be set to any value between 0 and STATCTRL[DACBFUP] by writing STATCTRL[DACBFRP].

36.5.1.1 DAC data buffer interrupts

There are several interrupts and associated flags that can be configured for the DAC buffer. STATCTRL[DACBFRPBF] is set when the DAC buffer read pointer reaches the DAC buffer upper limit, that is, STATCTRL[DACBFRP] = STATCTRL[DACBFUP]. STATCTRL[DACBFRPTF] is set when the DAC read pointer is equal to the start position, 0. Finally, STATCTRL[DACBFWMF] is set when the DAC buffer read pointer has reached the position defined by STATCTRL[DACBFWM]. STATCTRL[DACBFWM] can be used to generate an interrupt when the DAC buffer read pointer is between 1 to 4 words from STATCTRL[DACBFUP].

36.5.1.2 Modes of DAC data buffer operation

The following table describes the different modes of data buffer operation for the DAC module.

Table 36-34. Modes of DAC data buffer operation

Modes	Description
Buffer Normal mode	This is the default mode. The buffer works as a circular buffer. The read pointer increases by one, every time the trigger occurs. When the read pointer reaches the upper limit, it goes to 0 directly in the next trigger event.
Buffer Swing mode	This mode is similar to the normal mode. However, when the read pointer reaches the upper limit, it does not go to 0. It will descend by 1 in the next trigger events until 0 is reached.
Buffer One-time Scan mode	The read pointer increases by 1 every time the trigger occurs. When it reaches the upper limit, it stops there. If read pointer is reset to the address other than the upper limit, it will increase to the upper address and stop there again. **NOTE:** If the software set the read pointer to the upper limit, the read pointer will not advance in this mode.

36.5.2 DMA operation

When DMA is enabled, DMA requests are generated instead of interrupt requests. The DMA Done signal clears the DMA request.

The status register flags are still set and are cleared automatically when the DMA completes.

36.5.3 Resets

During reset, the DAC is configured in the default mode and is disabled.

36.5.4 Low-Power mode operation

The following table shows the wait mode and the stop mode operation of the DAC module.

Table 36-35. Modes of operation

Modes of operation	Description
Wait mode	The DAC will operate normally, if enabled.
Stop mode	If enabled, the DAC module continues to operate in Normal Stop mode and the output voltage will hold the value before stop. In low-power stop modes, the DAC is fully shut down.

NOTE

The assignment of module modes to core modes is chip-specific. For module-to-core mode assignments, see the chapter that describes how modules are configured.

Chapter 37 ADC-Digital-12b-1MSPS-SAR

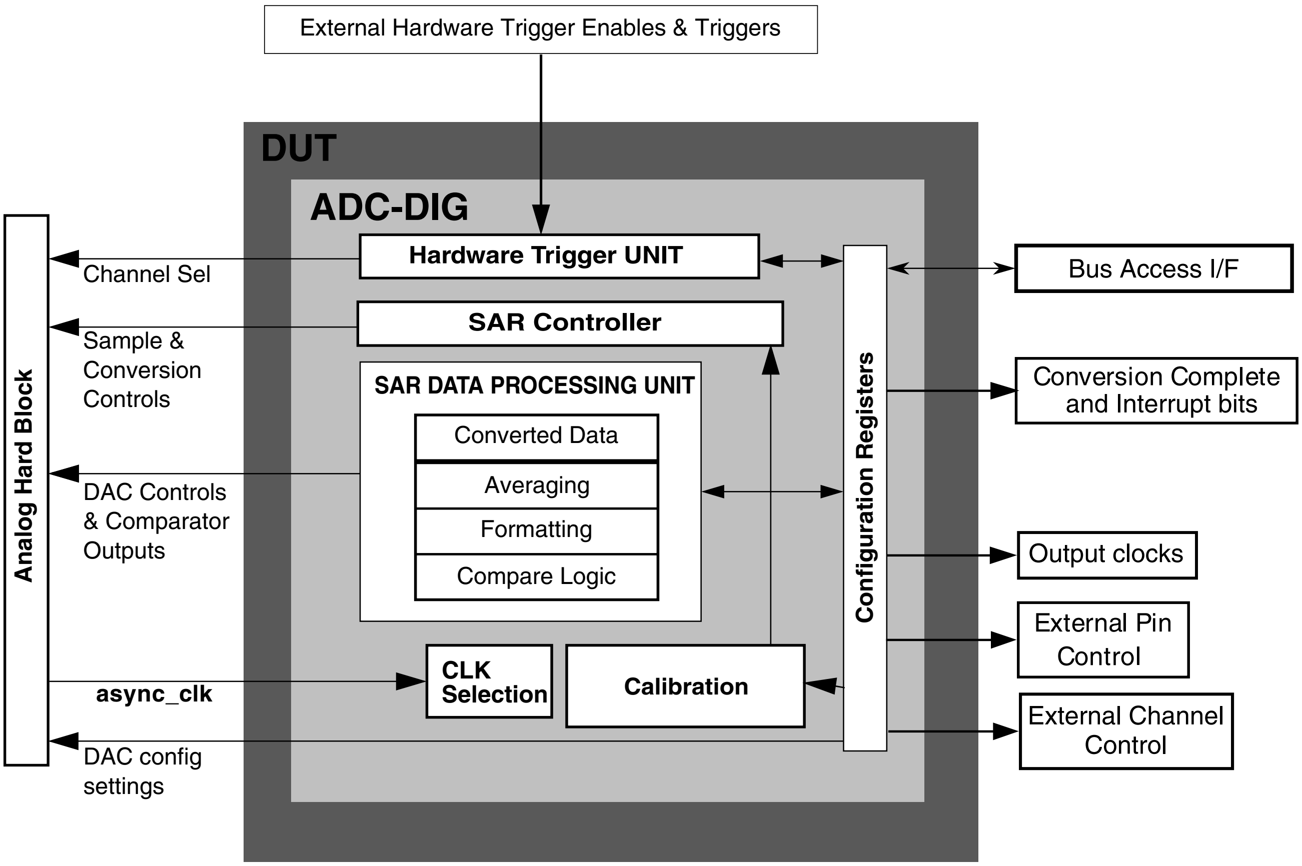
37.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

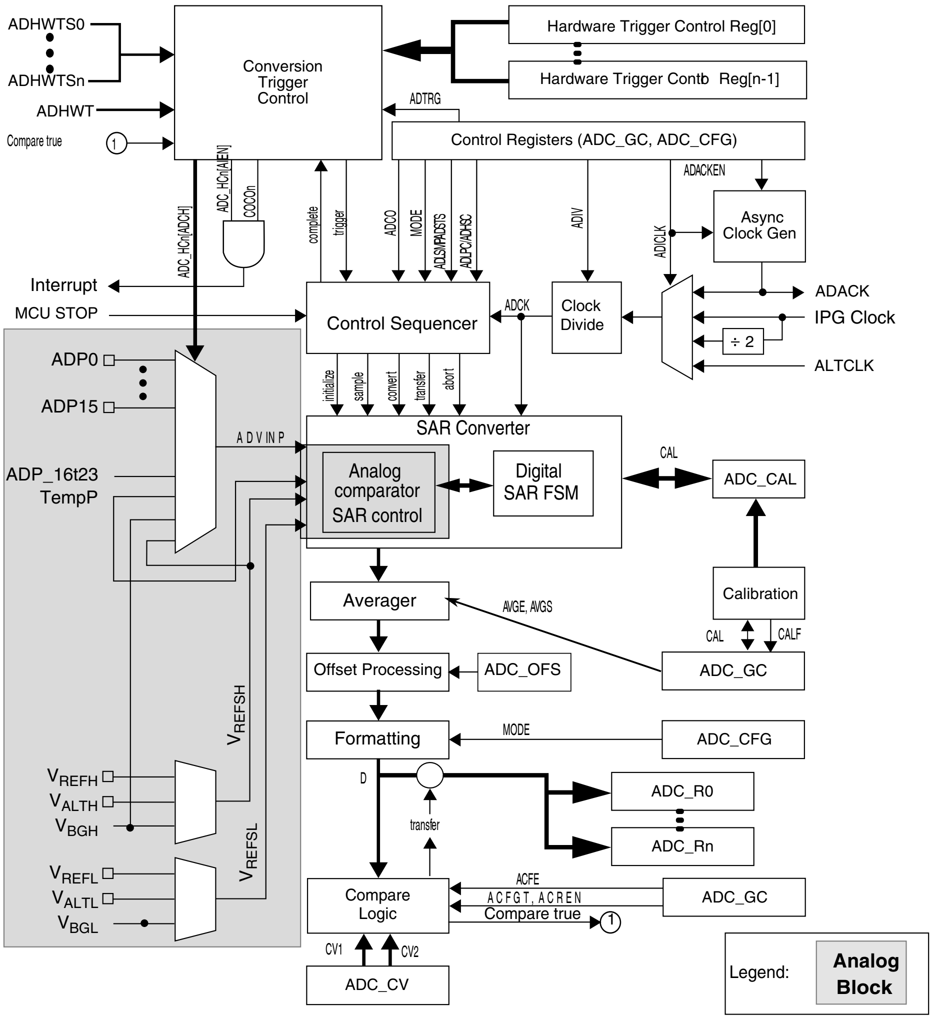
The 12-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

37.2 ADC-Digital I/F block diagram



37.2.1 ADC-Digital block diagram

The following figure shows a top-level block diagram of the ADC.


ADHWTS0
ADHWTSn
ADHWT
Compare true
Conversion Trigger Control
Hardware Trigger Control Reg[0]
Hardware Trigger Contb Reg[n-1]
ADTRG
Control Registers (ADC_GC, ADC_CFG)
ADACKEN
Async Clock Gen
ADC_HCn[ADCH]
ADC_HCn[AIEN]
COCOn
complete
trigger
ADCO
MODE
ADLSMP/ADSTS
ADLPC/ADHSC
ADIV
ADICLK
Interrupt
MCU STOP
Control Sequencer
ADCK
Clock Divide
ADACK
IPG Clock
÷ 2
ALTCLK
ADP0
ADP15
ADP_16t23
TempP
initialize
sample
convert
transfer
abort
ADVINP
SAR Converter
Analog comparator SAR control
Digital SAR FSM
CAL
ADC_CAL
Averager
AVGE, AVGS
Calibration
CAL
CALF
ADC_GC
Offset Processing
ADC_OFS
Formatting
MODE
ADC_CFG
VREFSH
VREFH
VALTH
VBGH
VREFL
VALTL
VBGL
VREFSL
D
transfer
ADC_R0
ADC_Rn
ACFE
ACFGT, ACREN
Compare true
Compare Logic
ADC_GC
CV1
CV2
ADC_CV
Legend:
Analog Block

37.3 Features List

The features of the ADC-Digital are as follows:

- Configuration registers
  - 32-bit, word aligned, byte enabled registers. (Byte and Halfword access is not supported)
- Linear successive approximation algorithm with up to 12-bit resolution with 10/11 bit accuracy.
- Up to 10 ENOB (dedicated Single Ended Channels)
- Up to 1MS/s sampling rate
- Up to 16 single-ended external analog inputs
- Single or continuous conversion (automatic return to idle after single conversion)
- Output Modes: (in right-justified unsigned format)
  - 12-bit,
  - 10-bit
  - 8-bit
- Configurable sample time and conversion speed/power
- Conversion complete and hardware average complete flag and interrupt
- Input clock selectable from up to four sources
- Asynchronous clock source for lower noise operation with option to output the clock
- Selectable asynchronous hardware conversion trigger with hardware channel select
- Selectable voltage reference, Internal, External, or Alternate
- Automatic compare with interrupt for less-than, greater-than or equal-to, within range, or out-of-range, programmable value
- Operation in low power modes for lower noise operation
- Temperature sensor
- Hardware average function
- Self-calibration mode

37.4 Operation mode (ADC enable/disable)

By default, the ADC is in disabled mode. In this state, no conversion or other actions occur. All of the ADC control registers are accessible in this state through an access bus interface.To enable the ADC, required configurations should be done by programming the ADC configuration registers.

37.5 ADC module interface

The ADC is connected to many interfaces such as the clocks and reset, access bus, voltage references, interrupt controller, hardware triggers, ADC pin control, and analog I/F as shown in the following figure.

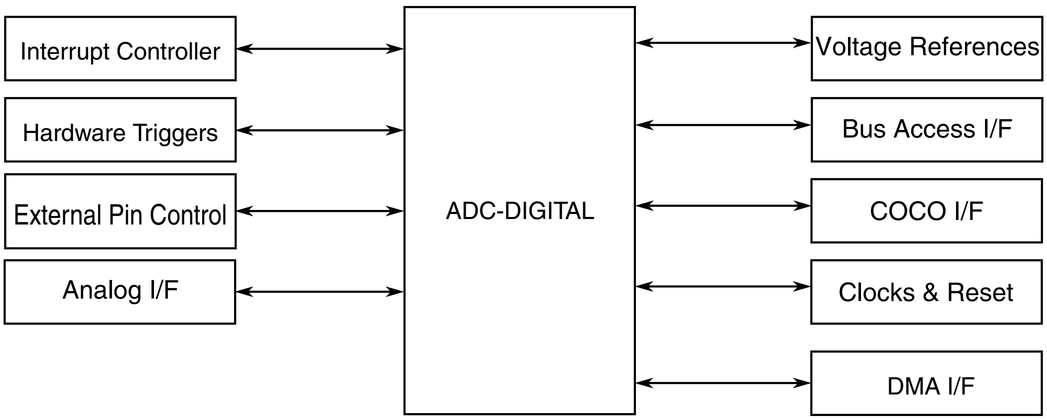


Figure 37-2. ADC-Digital module interface

The number of interrupt vectors and the trigger sources (modules that can trigger and ADC conversion) are chip-specific. Please refer to the Chip Configuration chapter for this information.

37.6 External Signal Description

The ADC module supports up to 16 single-ended inputs.

The ADC also requires four supply/reference/ground connections.

Name	Function
DADP0-DADP15	Analog Channel Inputs
ADP_16t23	Separate Analog Channel Inputs

Table continues on the next page...

Name	Function
V_{REFSH}	Voltage Reference Select High
V_{REFSL}	Voltage Reference Select Low
V_{DDAD}	Analog power supply
V_{SSAD}	Analog ground

37.7 Memory map and register definition

The ADC-Digital contains 32-bit, word aligned, byte enables registers; byte or half word access are not supported. All configuration registers are accessible via 32-bit access bus Interface. Write access to reserved locations have no impact while read access to reserved locations always return 0.

NOTE

No protection or indication mechanism is available (for example, 32-bit access starting with address offset value 0x01 or 0x02 or 0x03). The ADC does not check for correctness of the programmed values in the registers and the programmer must ensure that correct values are being written.

ADC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B000	Control register for hardware triggers (ADC0_HC0)	32	R/W	See section	37.7.1/1678
4003_B004	Control register for hardware triggers (ADC0_HC1)	32	R/W	See section	37.7.2/1679
4003_B008	Status register for HW triggers (ADC0_HS)	32	R (reads 0)	See section	37.7.3/1681
4003_B00C	Data result register for HW triggers (ADC0_R0)	32	R/W	See section	37.7.4/1682
4003_B010	Data result register for HW triggers (ADC0_R1)	32	R/W	See section	37.7.5/1683
4003_B014	Configuration register (ADC0_CFG)	32	R/W	See section	37.7.6/1684
4003_B018	General control register (ADC0_GC)	32	R/W	See section	37.7.7/1686
4003_B01C	General status register (ADC0_GS)	32	R/W	See section	37.7.8/1688
4003_B020	Compare value register (ADC0_CV)	32	R/W	See section	37.7.9/1689
4003_B024	Offset correction value register (ADC0_OFS)	32	R/W	See section	37.7.10/ 1690
4003_B028	Calibration value register (ADC0_CAL)	32	R/W	See section	37.7.11/ 1691
4003_B030	Pin control register (ADC0_PCTL)	32	R/W	See section	37.7.12/ 1691

ADC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_B000	Control register for hardware triggers (ADC1_HC0)	32	R/W	See section	37.7.1/1678
400B_B004	Control register for hardware triggers (ADC1_HC1)	32	R/W	See section	37.7.2/1679
400B_B008	Status register for HW triggers (ADC1_HS)	32	R (reads 0)	See section	37.7.3/1681
400B_B00C	Data result register for HW triggers (ADC1_R0)	32	R/W	See section	37.7.4/1682
400B_B010	Data result register for HW triggers (ADC1_R1)	32	R/W	See section	37.7.5/1683
400B_B014	Configuration register (ADC1_CFG)	32	R/W	See section	37.7.6/1684
400B_B018	General control register (ADC1_GC)	32	R/W	See section	37.7.7/1686
400B_B01C	General status register (ADC1_GS)	32	R/W	See section	37.7.8/1688
400B_B020	Compare value register (ADC1_CV)	32	R/W	See section	37.7.9/1689
400B_B024	Offset correction value register (ADC1_OFS)	32	R/W	See section	37.7.10/ 1690
400B_B028	Calibration value register (ADC1_CAL)	32	R/W	See section	37.7.11/ 1691
400B_B030	Pin control register (ADC1_PCTL)	32	R/W	See section	37.7.12/ 1691

37.7.1 Control register for hardware triggers (ADCx_HC0)

ADC_HC0 is for use only in hardware trigger mode. The ADC_HC0 to ADC_HCn(n=0,1) registers have identical fields, and are used to control ADC operation. At any one point in time, only one of the ADC_HC0 to ADC_HCn(n=0,1) registers is actively controlling ADC conversions. Updating ADC_HC0 while ADC_HCn(n=0,1) is actively controlling a conversion is allowed (and vice-versa for any of the ADC_HCn (n=0,1) registers). Writing ADC_HC0 while ADC_HC0 is actively controlling a conversion aborts the current conversion. In software trigger mode (ADTRG=0), writes to ADC_HC0 subsequently initiates a new conversion (if the ADCH bits are equal to a value other than all 1s). Similarly, writing any of the ADC_HCn(n=0,1) registers while that specific ADC_HC register is actively controlling a conversion aborts the current conversion. ADC_HC1 register is not used for software trigger operation and therefore writes to any of them do not initiate a new conversion.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								AIEN	Reserved		ADCH				
W																
Reset									0	0	0	1	1	1	1	1

ADCx_HC0 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 AIEN	Conversion Complete Interrupt Enable/Disable Control An interrupt is generated whenever ADC_HS[COCO0]=1 (conversion ADC_HC0 completed), provided the corresponding interrupt is enabled. 1 Conversion complete interrupt enabled 0 Conversion complete interrupt disabled
6–5 Reserved	This field is reserved.
4–0 ADCH	Input Channel Select This 5-bit field selects one of the input channels. The successive approximation converter subsystem is turned off when the channel select bits are all set (ADCH = 11111b). This feature allows for explicit

Table continues on the next page...

ADCx_HC0 field descriptions (continued)

Field	Description
	disabling of the ADC and isolation of the input channel from all sources. Terminating continuous conversions this way prevents an additional single conversion from being performed. 00000-01111 External channels 0 to 15. 10000-10111 8 external channels for satellite mux (external to this block) 11000 Reserved. 11001 VREFSH = internal channel, for ADC self-test, hard connected to VRH internally 11010 Reserved. 11011 Reserved. 11100-11110 Reserved. 11111 Conversion Disabled. Hardware Triggers will not initiate any conversion.

37.7.2 Control register for hardware triggers (ADCx_HC1)

ADC_HC1 are for use only in hardware trigger mode. The ADC_HC0 to ADC_HCn registers have identical fields, and are used to control ADC operation. At any one point in time, only one of the ADC_HC0 to ADC_HCn registers is actively controlling ADC conversions. Updating ADC_HC0 while ADC_HCn is actively controlling a conversion is allowed (and vice-versa for any of the ADC_HCn registers). Writing any of the ADC_HCn registers while that specific ADC_HCn register is actively controlling a conversion aborts the current conversion. Any of the ADC_HC1 - ADC_HCn registers are not used for software trigger operation and therefore writes to any of them do not initiate a new conversion.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								AIEN	0		ADCH				
W																
Reset									0	0	0	1	1	1	1	1

ADCx_HC1 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ADCx_HC1 field descriptions (continued)

Field	Description
7 AIEN	Conversion Complete Interrupt Enable/Disable Control An interrupt is generated whenever ADC_HS[COCO0]=1(conversion ADC_HC0 completed), provided the corresponding interrupt is enabled. 1 Conversion complete interrupt enabled 0 Conversion complete interrupt disabled
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 ADCH	Input Channel Select This 5-bit field selects one of the input channels. The successive approximation converter subsystem is turned off when the channel select bits are all set (ADCH =11111b). This feature allows for explicit disabling of the ADC and isolation of the input channel from all sources. Terminating continuous conversions this way prevents an additional single conversion from being performed. 00000-01111 External channels 0 to 15. 10000-10111 8 external channels for satellite mux (external to this block) 11000 Reserved. 11001 VREFSH = internal channel, for ADC self-test, hard connected to VRH internally 11010 Reserved. 11011 Reserved. 11100-11110 Reserved. 11111 Conversion Disabled. Hardware Triggers will not initiate any conversion.

37.7.3 Status register for HW triggers (ADCx_HS)

Bit 0 is used for both software and hardware trigger modes of operation. Bit 1 to bit (n-1) indicate the rest of the HW triggers' statuses similar to bit 0, potentially corresponding to multiple ADC_HC registers (for use only in hardware trigger mode).

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														COCO1	COCO0
W																
Reset															0	0

ADCx_HS field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 COCO1	See description for COCO0.
0 COCO0	Conversion Complete Flag The COCOn flag is a read-only bit that is set each time a conversion is completed when the compare function is disabled (ADC_GC[ACFE]=0) and the hardware average function is disabled (ADC_GC[AVGE]=0). When the compare function is enabled (ADC_GC[ACFE]=1), the COCOn flag is set upon completion of a conversion only if the compare result is true. When the hardware average function is enabled (ADC_GC[AVGE]=1), the COCOn flag is set upon completion of the selected number of conversions (determined by the ADC_CFG[AVGS] field). The COCO0 flag will also set at the completion of a Calibration and Test sequence. A COCOn bit is cleared when the respective ADC_HCn is written or when the respective ADC_Rn is read.

37.7.4 Data result register for HW triggers (ADCx_R0)

Contains the result of an ADC conversion of the channel selected by the respective hardware trigger and channel control register (ADC_HC0:ADC_HCn). For every ADC_HC0:ADC_HCn status and channel control register, there is a respective ADC_R0:ADC_Rn data result register. Unused bits in the ADC_Rn register are cleared in unsigned right justified modes. For example when configured for 10-bit single-ended mode, D[31:10] are cleared. The table below describes the behavior of the data result registers in the different modes of operation.

Table 37-21. Data Result Register Description

Conversion Mode	Data Result Register bits																Format
	D31	D30	....	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
12b single-ended	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	unsigned right justified
10b single-ended	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	unsigned right justified
8b single-ended	0	0	0	0	0	0	0	0	D	D	D	D	D	D	D	D	unsigned right justified

Address: Base address + Ch offset

Bit	31–12	11–0
R	0	D[11:0]
W		
Reset	0	0

ADCx_R0 field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 D[11:0]	Data (result of an ADC conversion)

37.7.5 Data result register for HW triggers (ADCx_R1)

Contains the result of an ADC conversion of the channel selected by the respective Hardware Trigger and channel control register (ADC_HC0:ADC_HCn). For every ADC_HC0:ADC_HCn status and channel control register, there is a respective ADC_R0 to ADC_Rn data result register. Unused bits in the ADC_Rn register are cleared in unsigned right justified modes. For example when configured for 10-bit single-ended mode, D[31:10] are cleared. The table below describes the behavior of the data result registers in the different modes of operation.

Table 37-23. Data Result Register Description

Conversion Mode	Data Result Register bits																Format
	D31	D30	....	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
12b single-ended	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	unsigned right justified
10b single-ended	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	unsigned right justified
8b single-ended	0	0	0	0	0	0	0	0	D	D	D	D	D	D	D	D	unsigned right justified

Address: Base address + 10h offset

Bit	31–12	11–0
R	0	D[11:0]
W		
Reset	0	0

ADCx_R1 field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 D[11:0]	Data (result of an ADC conversion)

37.7.6 Configuration register (ADCx_CFG)

Selects the mode of operation, clock source, clock divide, configure for low power, long sample time, high speed configuration and selects the sample time duration.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															OVWREN
W																
Reset	0															0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	AVGS		ADTRG	REFSEL		ADHSC	ADSTS		ADLPC	ADIV		ADLSMP	MODE		ADICLK	
Reset	0	0	0	0	0	0	1	0	0	0		0	0	0	0	0

ADCx_CFG field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 OVWREN	Data Overwrite Enable Controls the overwriting of the next converted Data onto the existing (previous) unread data into the Data result register. 1 Enable the overwritting. 0 Disable the overwritting. Existing Data in Data result register will not be overwritten by subsequent converted data.
15–14 AVGS	Hardware Average select Determines how many ADC conversions will be averaged to create the ADC average result. This functionality is activated when ADC_GC[AVGE] = 1. 00 4 samples averaged 01 8 samples averaged 10 16 samples averaged 11 32 samples averaged
13 ADTRG	Conversion Trigger Select Selects the type of trigger used for initiating a conversion. Two types of trigger are selectable: software trigger and hardware trigger. When software trigger is selected, a conversion is initiated following a write to ADC_HC0. When hardware trigger is selected, a conversion is initiated following the assertion of a pulse on Alternate Hardware trigger input along with the assertion of the enable of respective the hardware Triggers input .

Table continues on the next page...

ADCx_CFG field descriptions (continued)

Field	Description
	0 Software trigger selected 1 Hardware trigger selected
12–11 REFSEL	Voltage Reference Selection Selects the voltage reference source used for conversions. Refer to the chip configuration chapter's ADC section for details of these field definitions for this chip. 00 Selects VREFH/VREFL as reference voltage. 01 Selects VALTH/VALTL as reference voltage. 10 Selects VBGH/VBGL as reference voltage. 11 Reserved.
10 ADHSC	High Speed Configuration This bit configures the ADC for high speed operation. The internal ADC clock is higher than normal. 0 Normal conversion selected. 1 High speed conversion selected.
9–8 ADSTS	Defines the sample time duration. This has two modes, short and long. When long sample time is selected (ADLSMP=1) this works for long sample time otherwise this works for short sample. This allows higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption when continuous conversions are enabled if high conversion rates are not required. 00 Sample period (ADC clocks) = 2 if ADLSMP=0b Sample period (ADC clocks) = 12 if ADLSMP=1b 01 Sample period (ADC clocks) = 4 if ADLSMP=0b Sample period (ADC clocks) = 16 if ADLSMP=1b 10 Sample period (ADC clocks) = 6 if ADLSMP=0b Sample period (ADC clocks) = 20 if ADLSMP=1b 11 Sample period (ADC clocks) = 8 if ADLSMP=0b Sample period (ADC clocks) = 24 if ADLSMP=1b
7 ADLPC	Low-Power Configuration Puts the ADC hard block into low power mode and reduces the comparator enable period by controlling its timing in the SAR controller block towards the anlong hard block. The signal indicating low power mode to the Analog block is asserted when this bit is set. 0 ADC hard block not in low power mode. 1 ADC hard block in low power mode.
6–5 ADIV	Clock Divide Select Selects the divide ratio used by the ADC to generate the internal clock ADCK. 00 Input clock 01 Input clock / 2 10 Input clock / 4 11 Input clock / 8

Table continues on the next page...

ADCx_CFG field descriptions (continued)

Field	Description
4 ADLSMP	Long Sample Time Configuration Selects between different sample times based on the ADC_CFG[ADSTS] field. This bit adjusts the sample period to allow higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. If high conversion rates are not required, longer sample times can also be used to lower overall power consumption when continuous conversions are enabled. When ADLSMP=1, the Long Sample Time mode is selected and the time is defined by ADSTS[1:0] of the ADC_CFG register. 0 Short sample mode. 1 Long sample mode.
3–2 MODE	Conversion Mode Selection Used to set the ADC resolution mode. 00 8-bit conversion 01 10-bit conversion 10 12-bit conversion 11 Reserved
1–0 ADICLK	Input Clock Select Selects the input clock source to generate the internal clock ADCK. 00 IPG clock 01 IPG clock divided by 2 10 Alternate clock (ALTCLK) 11 Asynchronous clock (ADACK)

37.7.7 General control register (ADCx_GC)

Controls the calibration, continuous convert, hardware averaging functions, conversion active, hardware/software trigger select, compare function and voltage reference select of the ADC module.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CAL	ADCO	AVGE	ACFE	ACFGT	ACREN	DMAEN	ADACKEN
W																
Reset									0	0	0	0	0	0	0	0

ADCx_GC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CAL	Calibration CAL begins the calibration sequence when set. This bit stays set while the calibration is in progress and is cleared when the calibration sequence is complete. The ADC_GS[CALF] bit must be checked to determine the result of the calibration sequence. Once started, the calibration routine cannot be interrupted by writes to the ADC registers or the results will be invalid and the ADC_GS[CALF] bit will set. Setting the CAL bit will abort any current conversion.
6 ADCO	Continuous Conversion Enable Enables continuous conversions. 0 One conversion or one set of conversions if the hardware average function is enabled (AVGE=1) after initiating a conversion. 1 Continuous conversions or sets of conversions if the hardware average function is enabled (AVGE=1) after initiating a conversion.
5 AVGE	Hardware average enable Enables the hardware average function of the ADC. 0 Hardware average function disabled 1 Hardware average function enabled
4 ACFE	Compare Function Enable Enables the compare function. 0 Compare function disabled 1 Compare function enabled
3 ACFGT	Compare Function Greater Than Enable Configures the compare function to check the conversion result relative to the compare value register (ADC_CV) based upon the value of ACREN (bit 2 in ADC_GC register). The ACFE bit must be set for ACFGT to have any effect. 0 Configures "Less Than Threshold, Outside Range Not Inclusive and Inside Range Not Inclusive" functionality based on the values placed in the ADC_CV register. 1 Configures "Greater Than Or Equal To Threshold, Outside Range Inclusive and Inside Range Inclusive" functionality based on the values placed in the ADC_CV registers.
2 ACREN	Compare Function Range Enable Configures the compare function to check the conversion result of the input being monitored is either between or outside the range formed by the compare values in register (ADC_CV) determined by the value of ACFGT. The ACFE bit must be set for ACFGT to have any effect. 0 Range function disabled. Only the compare value 1 of ADC_CV register (CV1) is compared. 1 Range function enabled. Both compare values of ADC_CV registers (CV1 and CV2) are compared.
1 DMAEN	DMA Enable Enables the DMA logic.

Table continues on the next page...

ADCx_GC field descriptions (continued)

Field	Description
	0 DMA disabled (default) 1 DMA enabled
0 ADACKEN	Asynchronous clock output enable Enables the ADC's asynchronous clock source and the clock source output regardless of the conversion and input clock select (ADC_CFG[ADICLK]) settings of the ADC. Based on MCU configuration, the asynchronous clock may be used by other modules (see module introduction section). Setting this bit allows the clock to be used even while the ADC is idle or operating from a different clock source. Also, latency of initiating a single or first-continuous conversion with the asynchronous clock selected is reduced since the ADACK clock is already operational. 0 Asynchronous clock output disabled; Asynchronous clock only enabled if selected by ADICLK and a conversion is active. 1 Asynchronous clock and clock output enabled regardless of the state of the ADC

37.7.8 General status register (ADCx_GS)

Controls the calibration, continuous convert, hardware averaging functions, conversion active, hardware/software trigger select, compare function and voltage reference select of the ADC module.

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													AWKST	CALF	ADACT
W														w1c	w1c	
Reset														0	0	0

ADCx_GS field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 AWKST	Asynchronous wakeup interrupt status Holds the status of asynchronous interrupt status that occured during stop mode. This bit is set when ipg_stop is deasserted and ipg_clk has started. It is cleared by writing '1' to it. Clearing this bit also deasserts the Asynchronous interrupt to CPU. 1 Asynnchronous wake up interrupt occured in stop mode. 0 No asynchronous interrupt.
1 CALF	Calibration Failed Flag Displays the result of the calibration sequence. The calibration sequence will fail if Hardware Trigger is selected (i.e. ADC_CFG[ADTRG] = 1), or any ADC register is written, or any stop mode is entered before the calibration sequence completes. The CALF bit is cleared by writing a 1 to it. 0 Calibration completed normally. 1 Calibration failed. ADC accuracy specifications are not guaranteed.
0 ADACT	Conversion Active Indicates that a conversion or hardware averaging is in progress. ADACT is set when a conversion is initiated and cleared when a conversion is completed or aborted. 0 Conversion not in progress. 1 Conversion in progress.

37.7.9 Compare value register (ADCx_CV)

Contains compare values used to compare with the conversion result when the compare function is enabled (ADC_GC[ACFE]=1). The compare values are right justified as shown Figure . Therfore, the compare function only uses the compare value register bits that are related to the ADC mode of operation. (e.g. in 8 bit mode, CV1 = ADC_CV[7:0] and CV2 = ADC_CV[23:16], similarly in 10 bit mode, CV1 = ADC_CV[9:0] and CV2 = ADC_CV[25:16] etc.) The compare value 2 in this register is utilized only when the compare range function is enabled (ADC_GC[ACREN]=1).

Address: Base address + 20h offset

Bit	31–28	27–16	15–12	11–0
R	0	CV2	0	CV1
W				
Reset	0	0	0	0

ADCx_CV field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 CV2	Compare Value 2 Contains a compare value used to compare with the conversion result when the compare function and compare range function are enabled (ADC_GC[ACFE]=1, ADC_GC[ACREN]=1).
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 CV1	Compare Value 1 Contains a compare value used to compare with the conversion result when the compare function ais enabled (ADC_GC[ACFE]=1).

37.7.10 Offset correction value register (ADCx_OFS)

Contains the user-defined offset error correction value. This register is 13 bits wide. The value in the most significant bit (13th bit) is the operation bit. If this bit is ‘0’ then the value in the other 12 bits is added with the converted result value to generate final result to be loaded into ADC_Rn; if this bit is ‘1’ then this field is subtracted from converted value to generate final result (ADC_Rn).

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0															

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SIGN	OFS											
W																
Reset				0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_OFS field descriptions

Field	Description
31–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 SIGN	Sign bit 0 The offset value is added with the raw result 1 The offset value is subtracted from the raw converted value
11–0 OFS	Offset value User configurable offset value.

37.7.11 Calibration value register (ADCx_CAL)

Contains calibration information that is generated by the calibration function. This register contains a calibration value of four bits(CAL[3:0]); this is automatically set once the self calibration sequence is done (ADC_SC[CAL] bit is cleared). If this register is written to by the user after calibration, the linearity error specifications may not be met.

Address: Base address + 28h offset

Bit	31–4	3–0
R	0	CAL_CODE
W		
Reset	0	0

ADCx_CAL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 CAL_CODE	Calibration Result Value This value is automatically loaded and updated at the end of calibration.

37.7.12 Pin control register (ADCx_PCTL)

Disables the I/O port control of MCU pins used as analog inputs. ADC_PCTL can be used to control the pins associated with 24 channels (16 input + 8 external) of the ADC module.

Address: Base address + 30h offset

Bit	31–24	23	22	21	20	19	18	17	16
R	0	ADPC23	ADPC22	ADPC21	ADPC20	ADPC19	ADPC18	ADPC17	ADPC16
W									
Reset	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ADPC15	ADPC14	ADPC13	ADPC12	ADPC11	ADPC10	ADPC9	ADPC8	ADPC7	ADPC6	ADPC5	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_PCTL field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 ADPC23	ADC Pin Control 23 Controls the pin associated with channel AD23. 0 AD23 pin I/O control enabled 1 AD23 pin I/O control disabled
22 ADPC22	ADC Pin Control 22 Controls the pin associated with channel AD22. 0 AD22 pin I/O control enabled 1 AD22 pin I/O control disabled
21 ADPC21	ADC Pin Control 21 Controls the pin associated with channel AD21. 0 AD21 pin I/O control enabled 1 AD21 pin I/O control disabled
20 ADPC20	ADC Pin Control 20 Controls the pin associated with channel AD20. 0 AD20 pin I/O control enabled 1 AD20 pin I/O control disabled
19 ADPC19	ADC Pin Control 19 Controls the pin associated with channel AD19. 0 AD19 pin I/O control enabled 1 AD19 pin I/O control disabled
18 ADPC18	ADC Pin Control 18 Controls the pin associated with channel AD18. 0 AD18 pin I/O control enabled 1 AD18 pin I/O control disabled
17 ADPC17	ADC Pin Control 17 Controls the pin associated with channel AD17. 0 AD17 pin I/O control enabled 1 AD17 pin I/O control disabled
16 ADPC16	ADC Pin Control 16 Controls the pin associated with channel AD16. 0 AD16 pin I/O control enabled 1 AD16 pin I/O control disabled

Table continues on the next page...

ADCx_PCTL field descriptions (continued)

Field	Description
15 ADPC15	ADC Pin Control 15 Controls the pin associated with channel AD15. 0 AD15 pin I/O control enabled 1 AD15 pin I/O control disabled
14 ADPC14	ADC Pin Control 14 Controls the pin associated with channel AD14. 0 AD14 pin I/O control enabled 1 AD14 pin I/O control disabled
13 ADPC13	ADC Pin Control 13 Controls the pin associated with channel AD13. 0 AD13 pin I/O control enabled 1 AD13 pin I/O control disabled
12 ADPC12	ADC Pin Control 12 Controls the pin associated with channel AD12. 0 AD12 pin I/O control enabled 1 AD12 pin I/O control disabled
11 ADPC11	ADC Pin Control 11 Controls the pin associated with channel AD11. 0 AD11 pin I/O control enabled 1 AD11 pin I/O control disabled
10 ADPC10	ADC Pin Control 10 Controls the pin associated with channel AD10. 0 AD10 pin I/O control enabled 1 AD10 pin I/O control disabled
9 ADPC9	ADC Pin Control 9 Controls the pin associated with channel AD9. 0 AD9 pin I/O control enabled 1 AD9 pin I/O control disabled
8 ADPC8	ADC Pin Control 8 Controls the pin associated with channel AD8. 0 AD8 pin I/O control enabled 1 AD8 pin I/O control disabled
7 ADPC7	ADC Pin Control 7 Controls the pin associated with channel AD7.

Table continues on the next page...

ADCx_PCTL field descriptions (continued)

Field	Description
	0 AD7 pin I/O control enabled 1 AD7 pin I/O control disabled
6 ADPC6	ADC Pin Control 6 Controls the pin associated with channel AD6. 0 AD6 pin I/O control enabled 1 AD6 pin I/O control disabled
5 ADPC5	ADC Pin Control 5 Controls the pin associated with channel AD5. 0 AD5 pin I/O control enabled 1 AD5 pin I/O control disabled
4 ADPC4	ADC Pin Control 4 Controls the pin associated with channel AD4. 0 AD4 pin I/O control enabled 1 AD4 pin I/O control disabled
3 ADPC3	ADC Pin Control 3 Controls the pin associated with channel AD3. 0 AD3 pin I/O control enabled 1 AD3 pin I/O control disabled
2 ADPC2	ADC Pin Control 2 Controls the pin associated with channel AD2. 0 AD2 pin I/O control enabled 1 AD2 pin I/O control disabled
1 ADPC1	ADC Pin Control 1 Controls the pin associated with channel AD1. 0 AD1 pin I/O control enabled 1 AD1 pin I/O control disabled
0 ADPC0	ADC Pin Control 0 Controls the pin associated with channel AD0. 0 AD0 pin I/O control enabled 1 AD0 pin I/O control disabled

37.8 Functional Description

There are three possible states which ADC module can be in

1. Disabled State
2. Idle state
3. Performing conversions

Disabled State:

The ADC module is disabled during reset, stop mode (if internal clock is not selected as source of Clock), or when the ADCH bits of the hardware control (ADC_HCn) registers are all high.

Idle State:

The module is idle when a conversion has completed and another conversion has not been initiated. When idle and the asynchronous clock output enable is disabled (ADACKEN=0), the module is in its lowest power state.

Conversion State:

The ADC can perform an analog-to-digital conversion on any of the software selectable channels. All modes perform conversion by a successive approximation algorithm.

To meet accuracy specifications the ADC module must be calibrated using the on chip calibration function. Calibration is recommended to be done after any reset.

When the conversion is completed, the result is placed in the data result registers (ADC_Rn). The conversion complete flag (COCOn) bits in Hardware Status register is/are then set and an interrupt is generated, if the respective conversion complete interrupt has been enabled (ADC_HCn[AIEN]=1).

The ADC module has the capability of automatically comparing the result of a conversion with the contents of the compare value registers. The compare function is enabled by setting the ACFE (ADC compare function Enable) bit in ADC general control register.

The ADC module has the capability of automatically averaging the result of multiple conversions. The hardware average function is enabled by setting the AVGE bit in ADC general control register.

37.8.1 Clock Select and Divide Control

The ADC digital module has three clock sources

- IPG clock
- Alternate Clock (ALTCLK) is connected to the signal described in the ADC section of the chip configuration chapter.
- Internal clock (ADACK) is a dedicated clock used only by the ADC.

ADC digital block generates IPG clock/2 by internally dividing the IPG clock. The final clock is chosen from the following clocks.

- IPG clock
- IPG clock divided by 2
- ALTCLK
- ADACK

Out of the above four clocks one is chosen depending on the configuration of ADCLK[1:0] bits of ADC_CFG. This chosen clock is divided depending on the configuration of ADIV[1:0] bits of ADC_CFG. The final generated clock is used as conversion clock for ADC.

ADICLK	Selected Clock Source
00	IPG clock
01	IPG clock divided by 2
10	Alternate clock (ALTCLK)
11	Asynchronous clock (ADACK)

- The IPG clock. This is the default selection following reset.
- The IPG clock divided by two. For higher IPG clock rates, this allows a maximum divide by 16 of the IPG clock with using the ADIV bits.
- ALTCLK, as defined for this chip.
- The asynchronous clock (ADACK). This clock is generated from a clock source within the ADC module. Conversions are possible using ADACK as the input clock source while the MCU is in stop mode.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC may not perform according to specifications. If the available clocks are too fast, the clock must be divided to the appropriate frequency. This divider is specified by the ADIV bits and can be divide-by 1, 2, 4, or 8.

37.8.2 Input Select and Pin Control

The pin control registers (ADC_PC) disable the I/O port control of the pins used as analog inputs. When a pin control register bit is set, the following conditions are forced for the associated MCU pin:

- The output buffer is forced to its high impedance state.
- The input buffer is disabled. A read of the I/O port returns a zero for any pin with its input buffer disabled.
- The pullup is disabled.

37.8.3 Voltage Reference Selection

The ADC can be configured to accept one of three voltage reference pairs as the reference voltages (V_{REFSH} and $_{VREFSL}$) used for conversions. Each pair contains a positive reference which must be between the minimum Ref Voltage High and V_{DDAD}, and a ground reference which must be at the same potential as V_{SSAD}.The three pairs are external (V_{REFH} and V_{REFL}), alternate (V_{ALTH} and V_{ALTL}) and the internal bandgap (V_{BGH} and V_{BGL}). These voltage references are selected using the REFSEL bits in the ADC_CFG register.The alternate (V_{ALTH} and V_{ALTL}) voltage reference pair may select additional external pins or internal sources depending on MCU configuration.

37.8.4 Hardware Triggering and Channel Selection

The ADC module has a trigger input (known as alternate Trigger) which provides asynchronous hardware conversion trigger when the ADTRG bit in ADC configuration register (ADC_CFG) is set and any of the external hardware trigger select is high.

To be reliably captured, the Alternate trigger pulse must be high for sufficient time to satisfy clocking requirement of capturing Flop and the external hardware trigger select event must be set for sufficient time before and after the positive edge of Alternate trigger pulse to meet the setup / hold requirement of capturing flop.

If an external hardware trigger select event gets asserted during a conversion it must stay asserted until end of current conversion and remain set until the receipt of the an Alternate Trigger to initiate a new conversion.

When the Alternate trigger source is available and hardware triggering is enabled (ADC_CFG[ADTRG]=1), a conversion is initiated on the rising edge of the Alternate Trigger after a external hardware trigger select event has occurred.

If a conversion is in progress when a rising edge of a trigger occurs, the rising edge is ignored. In continuous convertion configuration, only the initial rising edge to launch continuous conversions is observed and until conversion gets aborted the ADC will continue to do conversions on the same ADC Hardware Trigger Control register that initiated the conversion. The hardware trigger function operates in conjunction with any of the conversion modes and configurations.

The channel selected for the conversion will depend on the settings of active Hardware Trigger register field ADC_HCn[ADCH] of enabled external hardware trigger.

NOTE

Asserting more than one external hardware trigger select signal at the same time will result in unknown results. To avoid this, only select one external hardware trigger select signal prior to the next intended conversion.

When the conversion is completed, the result is placed in the data registers associated with the external hardware trigger received (active trigger selects ADC_Rn). The conversion complete flag associated with the external hardware trigger received (ADC_HS[COCOn]) is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled (ADC_HCn[AIEN]=1).

37.8.5 Conversion Control

Conversions is performed as determined by the MODE bits in ADC_CFG register.

Conversions can be initiated by a either by a software trigger or hardware trigger. In addition, the ADC module can be configured for low power operation, long sample time, continuous conversion, hardware average and automatic compare of the conversion result to a software determined compare value.

37.8.5.1 Initiating Conversions

A conversion is initiated:

- Following a write to ADC_HC1 (with ADCHn bits not all 1's) and if software triggered operation is selected (ADTRG=0).

- Following a hardware trigger event if hardware triggered operation is selected (ADTRG=1) and a external hardware trigger select event has occurred. The channel selected will depend on the active trigger select signal active selects ADC_HC1; if neither is active the off condition is selected).

Note

Selecting more than one external hardware trigger select signal (ext_hwts[n]) prior to a conversion completion will generate unknown results. To avoid this, only select one hardware trigger select signal (ext_hwts[n]) prior to a conversion completion.

- Following the transfer of the result to the data registers when continuous conversion is enabled (ADCO=1 in ADC_GC register).

If continuous conversion is enabled, a new conversion is automatically initiated after the completion of the current conversion. In software triggered operation (ADTRG=0), continuous conversions begin after ADC_HC1 is written and continue until aborted. In hardware triggered operation(ADTRG=1 and one external hardware trigger select event has occurred), continuous conversions begin after a hardware trigger event and continue until aborted.

If hardware averaging is enabled, a new conversion is automatically initiated after the completion of the current conversion until the correct number of conversions is completed. In software triggered operation, conversions begin after ADC_HC1 is written. In hardware triggered operation, conversions begin after a hardware trigger. If continuous conversions is also enabled, a new set of conversions to be averaged are initiated following the last of the selected number of conversions.

37.8.5.2 Completing Conversions

A conversion is completed when the result of the conversion is transferred into the data result registers. (provided the compare function & hardware averaging is disabled), this is indicated by the setting of COCOn. If hardware averaging is enabled, COCOn sets only, if the last of the selected number of conversions is complete. If the compare function is enabled, COCOn sets and conversion result data is transferred only if the compare condition is true. If both hardware averaging and compare functions are enabled, then COCOn sets only if the last of the selected number of conversions is complete and the compare condition is true. An interrupt is generated, if ADC_HCn[AIEN] is high at the

time that COCOn is set and if DMAEN is set, DMA request is asserted, if COCOn is set. Both the requests get deasserted when COCOn is low, cleared, which happens when data is read.

In all modes a blocking mechanism prevents a new result from overwriting previous data in ADC_Rn, if the previous data is in the process of being read. When blocking is active (OVWREN=0 in ADC_CFG), the conversion result data transfer is blocked, COCOn is not set, and the new result is lost. In all other cases of operation, when a conversion result data transfer is blocked, another conversion is initiated regardless of the state of ADCO (single or continuous conversions enabled).

Note

If continuous conversions are enabled, the blocking mechanism could result in the loss of data occurring at specific timepoints. To avoid this issue, the data must be read in fewer cycles than an ADC conversion time, accounting for interrupt or software polling loop latency.

If single conversions are enabled, the blocking mechanism could result in several discarded conversions and excess power consumption. To avoid this issue, the data registers must not be read after initiating a single conversion until the conversion completes.

37.8.5.3 Aborting Conversions

Any conversion in progress is aborted when:

- The MCU enters stop mode with ADACK not enabled.
- In software trigger mode, write to ADC_HC0 register, while ADC_HC0 is actively (already) controlling a conversion, aborts the current conversion. Since, any of the ADC_HC1 - ADC_HCn registers are not used for software trigger operation therefore, write to any of them neither initiate a new conversion nor abort the software triggered active conversion.
- In hardware trigger mode, write to any one of the ADC_HC0 - ADC_HCn register, while that specific ADC_HC0 - ADC_HCn register is actively controlling a conversion, aborts the current conversion.
- A write to any ADC register besides the ADC_HC0: ADC_HCn registers occurs . This indicates a mode of operation change has occurred and the current conversion is therefore invalid.

Note

When a conversion is aborted, the contents of the data result registers, ADCRn are not altered. The data result registers continue to hold the values, transferred after the completion of the last successful conversion. If the conversion is aborted by a reset or stop (not operated with internal ADACK), ADCRn (data result register) return to their reset states.

37.8.5.4 Power Control

The ADC module remains in its idle state until a conversion is initiated. If ADACK is selected as the conversion clock source but the asynchronous clock output is disabled (ADACKEN=0), the ADACK clock generator will also remain in its idle state (disabled) until a conversion is initiated. If the asynchronous clock output is enabled (ADACKEN=1), it will remain active regardless of the state of the ADC or the MCU power mode.

Power consumption when the ADC is active can be reduced by setting ADLPC..

37.8.5.5 Sample Time and Total Conversion Time

The total conversion time depends upon the following:

- the sample phase time (as determined by ADLSMP and ADSTS bits in ADC_CFG register),
- the compare phase time (determined by MODE bits)
- the frequency of the conversion clock (fADCK).
- the MCU bus frequency (for Handshaking and selection of clock)

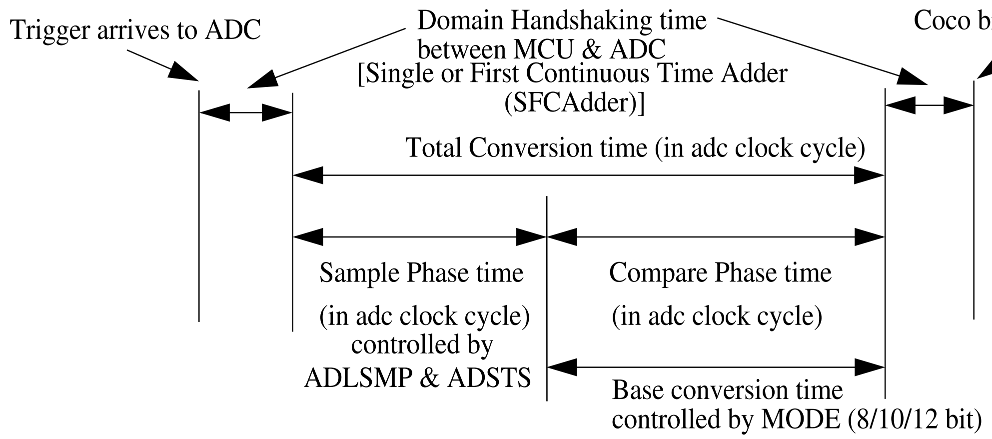


Figure 37-39. ADC conversion time details

After the module becomes active, sampling of the input begins. ADLSMP and ADSTS decide the sample time duration. When sampling is complete, the converter is isolated from the input channel and a successive approximation algorithm is performed to determine the digital value of the analog signal. The result of the conversion is transferred to ADC_Rn upon completion of the conversion algorithm.

If the bus frequency is less than the f_{ADCK} frequency, precise sample time for continuous conversions cannot be guaranteed .

The maximum total conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by the ADICLK bits in ADC_CFG register, and the divide ratio is specified by the ADIV bits.

The maximum total conversion time for all configurations is summarized in Figure 37-40. Refer to Table 37-46 through Table 37-49 for the variables referenced in the equation.

$$\text{ConversionTime} = \text{SFCAdder} + \text{AverageNum} \times (\text{BCT} + \text{LSTAdder})$$

Figure 37-40.

Table 37-46. Single or First Continuous Time Adder (SFCAdder)

ADACKEN	ADICLK	Single or First Continuous Time Adder (SFCAdder)
x	0x, 10	3 ADCK cycles (before starting of conversion) + 1ADCK (after end of conversion) + 2 bus clock cycles

Table continues on the next page...

Table 37-46. Single or First Continuous Time Adder (SFCAdder) (continued)

ADACKEN	ADICLK	Single or First Continuous Time Adder (SFCAdder)
1	11	3 ADCK cycles (before starting of conversion) + 1 ADCK (after end of conversion) + 2 bus clock cycles
0	11	1.5µs + 3 ADCK cycles (before starting of conversion) + 1 ADCK (after end of conversion) + 2 bus clock cycles

Table 37-47. Average Number Factor (AverageNum)

AVGE	AVGS[1:0]	Average Number Factor (AverageNum)
0	xx	1
1	00	4
1	01	8
1	10	16
1	11	32

Table 37-48. Base Conversion Time (BCT) (compare phase duration)

Mode	Base Conversion Time (BCT) (compare phase duration)
8 bit	17 ADCK cycles
10 bit	21 ADCK cycles
12 bit	25 ADCK cycles

Table 37-49. Long Sample Time

ADLSMP	ADSTS	Long Sample Time Adder (LSTAdder)
0	00	3 ADCK cycles
0	01	5 ADCK cycles
0	10	7 ADCK cycles (default)
0	11	9 ADCK cycles
1	00	13 ADCK cycles
1	01	17 ADCK cycles

Table continues on the next page...

Table 37-49. Long Sample Time (continued)

ADLSMP	ADSTS	Long Sample Time Adder (LSTAdder)
1	10	21 ADCK cycles
1	11	25 ADCK cycles

Note

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet ADC specifications.

37.8.5.6 Conversion Time Examples

The following examples uses Figure 37-40 and the information provided in tables Table 37-46 through Table 37-49.

37.8.5.6.1 Typical conversion time configuration

A typical configuration for ADC conversion is: 10-bit mode, with the bus clock selected as the input clock source, the input clock divide-by-1 ratio selected, and a bus frequency of 40 MHz, ADLSMP=0,ADLSTS=10 and high speed conversion disabled. The conversion time for a single conversion is calculated by using Figure 37-40 and the information provided in Table 37-50 through Table 37-52. The table below list the variables of Figure 37-40.

Table 37-50. Typical Conversion Time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	21 ADCK cycles
LSTAdder	7

The resulting conversion time is generated using the parameters listed in Table 37-50. So for Bus clock equal to 40 Mhz and ADCK equal to 40 Mhz the resulting conversion time is 0.95 us.

37.8.5.6.2 Long conversion time configuration

A configuration for long ADC conversion is: 12-bit mode, with the bus clock selected as the input clock source, the input clock divide-by-8 ratio selected, and a bus frequency of 40 MHz, long sample time enabled (ADLSMP=1, ADSTS=11) and configured for

longest adder and high speed conversion disabled. Average enabled for 32 conversions (AVGE=1, AVGS=11). The conversion time for this conversion is calculated by using equation on Sample Time and Total Conversion Time and the information provided in Table 37-46 through Table 37-49. The table below lists the variables of equation.

Table 37-51. Typical Conversion Time

Variable	Time
SFCAdder	3 ADCK cycles + 5 bus clock cycles
AverageNum	32
BCT	25 ADCK cycles
LSTAdder	25 ADCK cycles

The resulting conversion time is generated using the parameters listed in Table 37-51. So for Bus clock equal to 40 Mhz and ADCK equal to 5 Mhz the resulting conversion time is 10.0226 us (AverageNum). This results in a total conversion time of 320.725 us.

37.8.5.6.3 Short conversion time configuration

A configuration for short ADC conversion is: 8-bit mode, with the bus clock selected as the input clock source, the input clock divide-by-1 ratio selected, and a bus frequency of 40 MHz, long sample time disabled(ADLSMP=0, ADSTS=00) and high speed conversion enabled. The conversion time for this conversion is calculated by using the equation and the information provided in Table 37-46 to Table 37-49. The table below list the variables of equation.

Table 37-52. Typical Conversion Time

Variable	Time
SFCAdder	3 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	17 ADCK cycles
LSTAdder	3 ADCK cycles

The resulting conversion time is generated using the parameters listed in Table 37-52. So for Bus clock equal to 40Mhz and ADCK equal to 40Mhz the resulting conversion time is 700 ns.

37.8.5.7 Hardware Average Function

The hardware average function can be enabled (AVGE=1) to perform a hardware average of multiple conversions. The number of conversions is determined by the AVGS[1:0] bits, which select 4, 8, 16 or 32 conversions to be averaged. While the hardware average function is in progress the ADACT bit will be set.

After the selected input is sampled and converted, the result is placed in an accumulator from which an average is calculated once the selected number of conversions has been completed. When hardware averaging is selected the completion of a single conversion will not set the COCOn bit.

If the compare function is either disabled or evaluates true, after the selected number of conversions are completed, the average conversion result is transferred into the data result registers, ADC_Rn , and the COCOn bit is set. An ADC interrupt is generated upon the setting of COCOn if the respective ADC interrupt is enabled (AIENn=1).

37.8.6 Automatic Compare Function

The compare function can be configured to check if the result is less than or greater-than-or-equal-to a single compare value, or if the result falls within or outside a range determined by two compare values. The compare mode is determined by ACFGT, ACREN and the values in the compare value register (ADC_CV). After the input is sampled and converted, the compare values (CV1 and CV2) are used as described in the table below. There are six compare modes as shown in the table below.

Table 37-53. Compare Modes

ACFGT	ACREN	CV1 relative to CV2	Function	Compare Mode Description
0	0	-	Less than threshold	Compare true if the result is less than the CV1 registers.
1	0	-	Greater than or equal to threshold	Compare true if the result is greater than or equal to CV1 registers.
0	1	Less than or equal	Outside range, not inclusive	Compare true if the result is less than CV1 **Or** the result is Greater than CV2
0	1	Greater than	Inside range, not inclusive	Compare true if the result is less than CV1 **And** the result is greater than CV2
1	1	Less Than or equal	Inside range, inclusive	Compare true if the result is greater than or equal to CV1 **And** the result is less than or equal to CV2
1	1	Greater than	Outside range, inclusive	Compare true if the result is greater than or equal to CV1 **Or** the result is less than or equal to CV2

With the ADC range enable bit set, ADCREN =1, if compare value 1(CV1 value) is less than or equal to the compare value 2 (CV2 value), setting ACFGT will select a trigger-if-inside-compare-range, inclusive-of-endpoints function. Clearing ACFGT will select a trigger-if-outside-compare-range, not-inclusive-of-endpoints function.

If CV1 is greater than the CV2, setting ACFGT will select a trigger-if-outside-compare-range, inclusive-of-endpoints function. Clearing ACFGT will select a trigger-if-inside-compare-range, not-inclusive-of-endpoints function.

If the condition selected evaluates true, COCOn is set.

Upon completion of a conversion while the compare function is enabled, if the compare condition is not true, COCOn is not set and the conversion result data will not be transferred to the result register. If the hardware averaging function is enabled, the compare function compares the averaged result to the compare values. The same compare function definitions apply. An ADC interrupt is generated upon the setting of COCOn if the respective ADC interrupt is enabled (ADC_HCn[AIEN]=1).

Note

The compare function can monitor the voltage on a channel while the MCU is in wait or stop3 mode. The ADC interrupt wakes the MCU when the compare condition is met.

37.8.7 Calibration Function

The ADC contains a self-calibration function that is required to achieve the specified accuracy. Calibration should be run or valid calibration values should be written after power up and system reset (as the calibration register will be reset on reset assertion) with specified settings before any conversion is initiated. The calibration function sets the calibration value at the end of running the full calibration sequence in ADC_CAL register. The user must configure the ADC correctly prior to starting the calibration process, and must allow the process to run the full calibration sequence by checking the status of ADC_GC[CAL] and ADC_GS[CALF] so that the generated calibration value can be loaded.

Prior to calibration, the user must configure the ADC's clock source and frequency, low power configuration, voltage reference selection, sample time, averaging, and the high speed configuration according to the application's clock source availability and needs. If the application uses the ADC in a wide variety of configurations, the configuration for which the highest accuracy is required should be selected, or multiple calibrations can be done for the different configurations. The input channel, conversion mode, continuous function and compare function are all ignored during the calibration process.

To initiate calibration, the user sets the CAL bit and the calibration will automatically begin if the ADTRG bit = 0. If ADTRG = 1, the CAL bit will not get set and the calibration fail flag (CALF) will be set. While calibration is active, no ADC register can be written and no stop mode may be entered or the calibration routine will be aborted causing the CAL bit to clear and the CALF bit to set.

At the end of a calibration sequence the COCO[0] bit of the ADC_HS register will be set. The ADC_HCn[AIEN] bit can be used to allow an interrupt to occur at the end of a calibration sequence. If, at the end of calibration routine, the CALF bit is not set, the automatic calibration routine completed successfully.

To complete calibration, the user must follow the below procedure :

- Configure ADC_CFG with actual operating values for maximum accuracy.
- Configure ADC_TEST register with all zero
- Configure the ADC_GC values along with CAL bit
- Check the status of CALF bit in ADC_GS and the CAL bit in ADC_GC
- When CAL bit becomes '0' then check the CALF status and COCO[0] bit status

When complete the user may reconfigure and use the ADC as desired.

A second calibration may also be performed if desired by clearing and again setting the CAL bit

Overall the calibration routine may take as many as 14000 ADCK cycles and 100 bus cycles, depending on the results and the clock source chosen.

37.8.8 User Defined Offset Function

The ADC Offset Correction Register (ADC_OFS) contains the user configured offset value. This register is 13 bit wide. The value in MSB (13th bit) is the operation bit, if this bit is ‘0’ then the value in rest 12 bit is added with the converted result value to generate final result to be loaded into ADC_Rn and if this bit is ‘1’ then this field is subtracted from converted value to generate final Result (ADC_Rn). If the Final result is above the maximum or below the minimum result value, it is forced to the appropriate limit for the current mode of operation. Forced to 0x0FFF if over and 0x0000 if lower for 12 bit mode.

The offset value has no effect during calibration and BIST test on final result.

The formatting of the ADC Offset Register is different from the Data Result Registers (ADC_Rn) to preserve the resolution of the value regardless of the conversion mode selected. Lower order bits are ignored in lower resolution modes. For example, in 8b single-ended mode, the bits OFS[11:4] are subtracted from D[7:0] when bit OFS[12] (sign bit) is ‘1’ ; indicates subtraction and bits OFS[4:0] are ignored. For 12b single-ended mode, bits OFS[11:0] are directly subtracted from the conversion result data D[11:0] when OFS[12] (sign bit) is ‘1’. The similar is the addition operation when OFS[12](sign bit) is 0.

ADC_OFS is manually set according to user requirements once the self calibration sequence is done (CAL is cleared). The user have to write ADC_OFS with desired value.

NOTE

There is an effective limit to the values of Offset that can be set by the user. If the magnitude of the offset is too great the results of the conversions will cap off at the limits.

The offset function may be employed by the user to remove application offsets or DC bias values. An offset correction that results in an out-of-range value will be forced to the minimum or maximum value.

For applications which may change the offset repeatedly during operation, it is recommended to store the initial offset value in flash so that it can be recovered and added to any user offset adjustment value and the sum stored in the ADC_OFS registers.

37.8.9 Temperature Sensor

The ADC module includes a temperature sensor whose output is connected to one of the ADC analog channel inputs (ADC_HCn[AIEN] = 011010 = 26). This channel can be selected through configuring the ADCHn bits of any ADC_HCn register. Any trigger on this will give a temperature value converted for further processings.

Figure 37-40 provides an approximate transfer function of the temperature sensor .

$$\mathrm{Temp} = 25 - \left(\left(V_{\mathrm{TEMP}} - V_{\mathrm{TEMP25}}\right) \div m\right)$$

where:

- V_{TEMP} is the voltage of the temperature sensor channel at the ambient temperature.
- V_{TEMP25} is the voltage of the temperature sensor channel at 25°C.
- m is the hot or cold voltage versus temperature slope in V/°C.

For temperature calculations, use the V_{TEMP25} and m values from the ADC Electricals table.

In application code, the user reads the temperature sensor channel, calculates V_{TEMP}, and compares to V_{TEMP25}. If V_{TEMP} is greater than V_{TEMP25} the cold slope value is applied in Figure 37-40. If V_{TEMP} is less than V_{TEMP25} the hot slope value is applied in Figure 37-40.

For more information on using the temperature sensor, consult AN3031.

37.8.10 MCU Wait Mode Operation

Wait mode is a **lower power-consumption standby mode** from which **recovery is fast** because **the clock sources remain active**. If a conversion is in progress when the MCU enters wait mode, it continues until completion. Conversions can be initiated while the MCU is in wait mode by means of the hardware trigger or if continuous conversions are enabled.

The bus clock, bus clock divided by two, and ADACK are available as conversion clock sources while in wait mode. The use of ALTCLK as the conversion clock source in wait is dependent on the definition of ALTCLK for the specific MCU.

A conversion complete event sets the COCOn and generates an ADC interrupt to wake the MCU from wait mode.

If the compare and hardware averaging functions are disabled, a conversion complete event sets the COCOn and generates an ADC interrupt to wake the MCU from wait mode if the respective ADC interrupt is enabled (ADC_HCn[AIEN]=1).

If the hardware averaging function is enabled the COCOn will set (and generate an interrupt if enabled) when the selected number of conversions are complete.

If the compare function is enabled the COCOn will set (and generate an interrupt if enabled) only if the compare conditions are met.

If a single conversion is selected and the compare trigger is not met, the ADC will return to its idle state and cannot wake the MCU from wait mode unless a new conversion is initiated by the hardware trigger.

37.8.11 MCU Stop Mode Operation

Stop mode is a low power-consumption standby mode during which most or all clock sources on the MCU are disabled. Stop mode entered when stop indication comes from the MCU.

37.8.11.1 Stop Mode With ADACK Disabled

If the asynchronous clock, ADACK, is not selected as the conversion clock, executing a stop instruction aborts the current conversion and places the ADC in its idle state. The contents of the ADC registers, including ADC_Rn are unaffected by stop mode. After exiting from stop mode, a software or hardware trigger is required to resume conversions.

37.8.11.2 Stop Mode With ADACK Enabled

If ADACK is selected as the conversion clock, the ADC continues operation during stop mode. For guaranteed ADC operation, the MCU's voltage regulator must remain active during stop mode.

If a conversion is in progress when the MCU enters stop mode, it continues until completion. Conversions can be initiated while the MCU is in stop mode by means of the hardware trigger or if continuous conversions are enabled.

A conversion complete event sets the COCOn and generates an ADC interrupt to wake the MCU from stop mode :

Note

The ADC module can wake the system from low-power stop and cause the MCU to begin consuming run-level currents without generating a system level interrupt. To prevent this scenario, software should ensure the conversion result data transfer blocking mechanism (discussed in Completing Conversions) is cleared when entering stop and continuing ADC conversions.

37.9 Initialization Information

This section gives an example that provides some basic direction on how to initialize and configure the ADC module. User can configure the module for 8, 10, 12 bit resolution, single or continuous conversion, and a polled or interrupt approach, among many other options.

37.9.1 ADC Module Initialization Example

This section describes the initialization sequence along with pseudo-code.

37.9.1.1 Initialization Sequence

Before the ADC module can be used to complete conversions, an initialization procedure must be performed. A typical sequence is as follows:

- Calibrate the ADC by following the calibration instructions in Calibration Function
- Update the configuration register (ADC_CFG) to select the input clock source and the divide ratio used to generate the internal clock, ADCK. This register is also used for selecting sample time and low-power configuration.
- Update Genral control register (ADC_GC) to select whether conversions will be continuous or completed only once (ADCO) and to select whether to perform hardware averaging, etc.
- Update Trigger control register (ADC_HCn) to select the conversion trigger (hardware or software, i.e. configure ADTRG bit) and compare function options, if enabled.

37.9.1.2 Pseudo-Code Example

In this example, the ADC module is set up with interrupts enabled to perform a single 10-bit conversion at low power with a long sample time on input channel 1, where the internal ADCK clock is derived from the bus clock divided by 1.

ADC_CFG

```
Bit 7    ADLPC    1      Configures for low power (lowers maximum clock speed.
```

```
Bit 6:5  ADIV    00     Sets the ADCK to the input clock  ÷  1.
Bit 4    ADLSMP  1      Configures for long sample time.
Bit 3:2  MODE    10     Sets mode at 10-bit conversions.
Bit 1:0  ADICLK  00     Selects bus clock as input clock source.
```

ADC_GC

```
Bit 7    CAL     0      Flag indicates if a conversion is in progress.
Bit 6    ADCO    0      Software trigger selected.
Bit 5    AVGE    0      Compare function disabled.
Bit 4    ACFE    0      Compare function disabled.
Bit 3    ACFGT   0      Not used in this example.
Bit 2    ACREN   0      Not used in this example.
Bit 1    DMAEN   0      Not used in this example.
Bit 0    ADACKEN 0      Not used in this example.
```

ADC_HC0

```
   Bit 7    AIEN   1      Conversion complete interrupt enabled.
Bit 4:0  ADCH   00001  Input channel 1 selected as ADC input channel.
```

ADC_R0

Holds results of conversion. Read high byte (ADCRHA) before low byte (ADCRLA) so that conversion data cannot be overwritten with data from the next conversion.

ADC_CV

Holds compare values when compare function enabled.

ADC_PCTL

AD1 pin I/O control disabled. All other AD pins remain general purpose I/O pins.

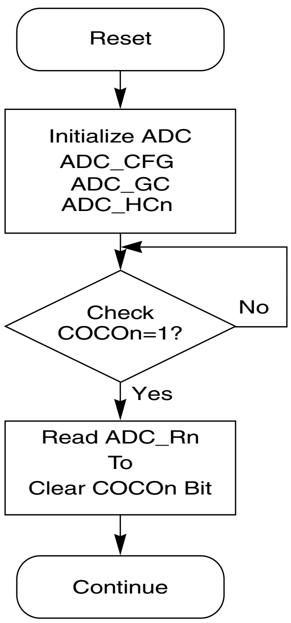


Figure 37-41. Initialization Flowchart for Example

37.10 Application Information

This section contains information for using the ADC module in applications. The ADC has been designed to be integrated into a microcontroller for use in embedded control applications requiring an A/D converter.

37.10.1 Sources of Error

Several sources of error exist for A/D conversions. These are discussed in the following sections.

37.10.1.1 Sampling Error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy. Given the maximum input resistance of approximately 7kΩ and input capacitance of approximately 1.3 pF, sampling to within 1/4LSB (at 12-bit resolution) can be achieved within the nominal sample window (6 cycles @ 40 MHz maximum ADCK frequency) provided the resistance of the external analog source (R_{AS}) is kept below 4 kΩ.

Higher source resistances or higher-accuracy sampling is possible by setting ADLSMP and changing the ADSTS bits (to increase the sample window) or decreasing ADCK frequency to increase sample time.

37.10.1.2 Pin Leakage Error

Leakage on the I/O pins can cause conversion error if the external analog source resistance (R_{AS}) is high. If this error cannot be tolerated by the application, keep R_{AS} lower than V_{DDAD} / (2^N*I_{LEAK}) for less than 1/4LSB leakage error (N = 8 in 8-bit, 10 in 10-bit or 12 in 12-bit mode).

37.10.1.3 Noise-Induced Errors

System noise that occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is a 0.1 μF low-ESR capacitor from V_{REFH} to V_{REFL}.
- There is a 0.1 μF low-ESR capacitor from V_{DDAD} to V_{SSAD}.
- If inductive isolation is used from the primary supply, an additional 1 μF capacitor is placed from V_{DDAD} to V_{SSAD}.
- V_{SSAD} (and V_{REFL}, if connected) is connected to V_{SS} at a quiet point in the ground plane.
- Operate the MCU in wait or stop mode before initiating (hardware triggered conversions) or immediately after initiating (hardware or software triggered conversions) the ADC conversion.
  - For software triggered conversions, immediately follow the write to ADCSC1 with a wait instruction or stop instruction.
  - For stop mode operation, select ADACK as the clock source. Operation in stop reduces V_{DD} noise but increases effective conversion time due to stop recovery.
- There is no I/O switching, input or output, on the MCU during the conversion.

There are some situations where external system activity causes radiated or conducted noise emissions or excessive V_{DD} noise is coupled into the ADC. In these situations, or when the MCU cannot be placed in wait or stop or I/O activity cannot be halted, these recommended actions may reduce the effect of noise on the accuracy:

- Place a 0.01 μF capacitor (CAS) on the selected input channel to V_{REFL} or V_{SSAD} (this improves noise issues, but affects the sample rate based on the external analog source resistance).
- Average the result by converting the analog input many times in succession and dividing the sum of the results. Four samples are required to eliminate the effect of a 1LSB, one-time error.
- Reduce the effect of synchronous noise by operating off the asynchronous clock (ADACK) and averaging. Noise that is synchronous to ADCK cannot be averaged out.

37.10.1.4 Code Width and Quantization Error

Note

This will remain the same as long as the result is rounded for 8 and 10-bit modes. If the result is truncated in 8/10b modes then they will match 12b mode where the quantization error is -1 to 0

The ADC quantizes the ideal straight-line transfer function into 4096 steps (in 12-bit mode). Each step ideally has the same height (1 code) and width. The width is defined as the delta between the transition points to one code and the next. The ideal code width for an N bit converter (in this case N can be 8, 10 or 12), defined as 1LSB, is:

$$1\text{lsb} = (V_{\text{REFH}} - V_{\text{REFL}}) / 2^N$$

There is an inherent quantization error due to the digitization of the result. For 8-bit or 10-bit conversions the code transitions when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be ± 1/2 lsb in 8- or 10-bit mode. As a consequence, however, the code width of the first (0x000) conversion is only 1/2 lsb and the code width of the last (0xFF or 0x3FF) is 1.5 lsb.

For 12-bit conversions the code transitions only after the full code width is present, so the quantization error is -1 lsb to 0 lsb and the code width of each step is 1 lsb.

37.10.1.5 Linearity Errors

The ADC may also exhibit non-linearity of several forms. Every effort has been made to reduce these errors but the system should be aware of them because they affect overall accuracy. These errors are:

- Zero-scale error (E_{ZS}) (sometimes called offset) — This error is defined as the difference between the actual code width of the first conversion and the ideal code width (1/2 lsb in 8-bit or 10-bit modes and 1 lsb in 12-bit mode). If the first conversion is 0x001, the difference between the actual 0x001 code width and its ideal (1 lsb) is used.
- Full-scale error (E_{FS}) — This error is defined as the difference between the actual code width of the last conversion and the ideal code width (1.5 lsb in 8-bit or 10-bit modes and 1LSB in 12-bit mode). If the last conversion is 0x3FE, the difference between the actual 0x3FE code width and its ideal (1LSB) is used.

- Differential non-linearity (DNL) — This error is defined as the worst-case difference between the actual code width and the ideal code width for all conversions.
- Integral non-linearity (INL) — This error is defined as the highest-value the (absolute value of the) running sum of DNL achieves. More simply, this is the worst-case difference of the actual transition voltage to a given code and its corresponding ideal transition voltage, for all codes.
- Total unadjusted error (TUE) — This error is defined as the difference between the actual transfer function and the ideal straight-line transfer function and includes all forms of error.

37.10.1.6 Code Jitter, Non-Monotonicity, and Missing Codes

Analog-to-digital converters are susceptible to three special forms of error. These are code jitter, non-monotonicity, and missing codes.

Code jitter is when, at certain points, a given input voltage converts to one of two values when sampled repeatedly. Ideally, when the input voltage is infinitesimally smaller than the transition voltage, the converter yields the lower code (and vice-versa). However, even small amounts of system noise can cause the converter to be indeterminate (between two codes) for a range of input voltages around the transition voltage. This range is normally around ±1/2 lsb in 8-bit or 10-bit mode, or around 2 lsb in 12-bit mode, and increases with noise.

This error may be reduced by repeatedly sampling the input and averaging the result. Additionally the techniques discussed in Noise-Induced Errors reduces this error.

Non-monotonicity is defined as when, except for code jitter, the converter converts to a lower code for a higher input voltage. Missing codes are those values never converted for any input value.

In 8-bit or 10-bit mode, the ADC is guaranteed to be monotonic and have no missing codes.

Chapter 38
Programmable Delay Block (PDB)

38.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The Programmable Delay Block (PDB) provides controllable delays from either an internal or an external trigger, or a programmable interval tick, to the hardware trigger inputs of ADCs and/or generates the interval triggers to DACs, so that the precise timing between ADC conversions and/or DAC updates can be achieved.

38.1.1 Features

- Up to 15 trigger input sources and one software trigger source
- Up to eight configurable PDB channels for ADC hardware trigger
    - One PDB channel is associated with one ADC
    - One trigger output for ADC hardware trigger and up to eight pre-trigger outputs for ADC trigger select per PDB channel
    - Trigger outputs can be enabled or disabled independently
    - One 16-bit delay register per pre-trigger output
    - Optional bypass of the delay registers of the pre-trigger outputs
    - Operation in One-Shot or Continuous modes
    - Optional back-to-back mode operation, which enables the ADC conversions complete to trigger the next PDB channel

  - One programmable delay interrupt
  - One sequence error interrupt
  - One channel flag and one sequence error flag per pre-trigger
  - DMA support
- Up to eight DAC interval triggers
  - One interval trigger output per DAC
  - One 16-bit delay interval register per DAC trigger output
  - Optional bypass of the delay interval trigger registers
  - Optional external triggers
- Up to eight pulse outputs (pulse-out's)
  - Pulse-out's can be enabled or disabled independently
  - Programmable pulse width

NOTE

The number of PDB input and output triggers are chip-specific. See the chip configuration information for details.

38.1.2 Implementation

In this section, the following letters refer to the number of output triggers:

- N—Total available number of PDB channels.
- n—PDB channel number, valid from 0 to N-1.
- M—Total available pre-trigger per PDB channel.
- m—Pre-trigger number, valid from 0 to M-1.
- X—Total number of DAC interval triggers.
- x—DAC interval trigger output number, valid from 0 to X-1.
- Y—Total number of Pulse-Out's.
- y—Pulse-Out number, valid value is from 0 to Y-1.

NOTE

The number of module output triggers to core is chip-specific. For module to core output triggers implementation, see the chip configuration information.

38.1.3 Back-to-back acknowledgment connections

PDB back-to-back operation acknowledgment connections are chip-specific. For implementation, see the chip configuration information.

38.1.4 Block diagram

This diagram illustrates the major components of the PDB.

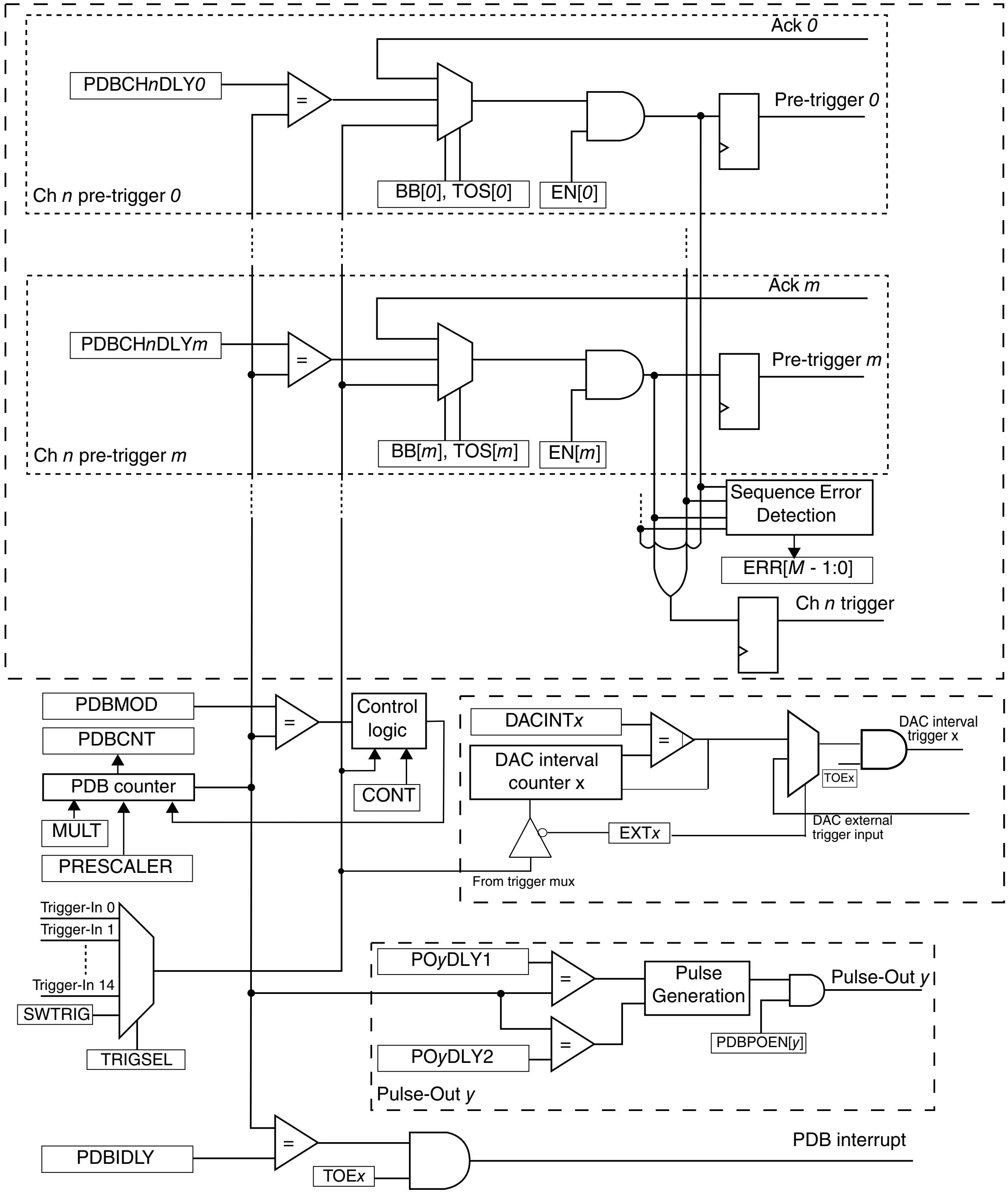


Figure 38-1. PDB block diagram

In this diagram, only one PDB channel *n*, one DAC interval trigger *x*, and one Pulse-Out *y* are shown. The PDB-enabled control logic and the sequence error interrupt logic are not shown.

38.1.5 Modes of operation

PDB ADC trigger operates in the following modes:

- Disabled—Counter is off, all pre-trigger and trigger outputs are low if PDB is not in back-to-back operation of Bypass mode.
- Debug—Counter is paused when processor is in Debug mode, and the counter for the DAC trigger is also paused in Debug mode.
- Enabled One-Shot—Counter is enabled and restarted at count zero upon receiving a positive edge on the selected trigger input source or software trigger is selected and SC[SWTRIG] is written with 1. In each PDB channel, an enabled pre-trigger asserts once per trigger input event. The trigger output asserts whenever any of the pre-triggers is asserted.
- Enabled Continuous—Counter is enabled and restarted at count zero. The counter is rolled over to zero again when the count reaches the value specified in the modulus register, and the counting is restarted. This enables a continuous stream of pre-triggers/trigger outputs as a result of a single trigger input event.
- Enabled Bypassed—The pre-trigger and trigger outputs assert immediately after a positive edge on the selected trigger input source or software trigger is selected and SC[SWTRIG] is written with 1, that is the delay registers are bypassed. It is possible to bypass any one or more of the delay registers; therefore, this mode can be used in conjunction with One-Shot or Continuous mode.

38.2 PDB signal descriptions

This table shows the detailed description of the external signal.

Table 38-1. PDB signal descriptions

Signal	Description	I/O
EXTRG	External Trigger Input Source If the PDB is enabled and external trigger input source is selected, a positive edge on the EXTRG signal resets and starts the counter.	I

38.3 Memory map and register definition

PDB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_6000	Status and Control register (PDB_SC)	32	R/W	0000_0000h	38.3.1/1725
4003_6004	Modulus register (PDB_MOD)	32	R/W	0000_FFFFh	38.3.2/1727
4003_6008	Counter register (PDB_CNT)	32	R	0000_0000h	38.3.3/1728
4003_600C	Interrupt Delay register (PDB_IDLY)	32	R/W	0000_FFFFh	38.3.4/1728
4003_6010	Channel n Control register 1 (PDB_CH0C1)	32	R/W	0000_0000h	38.3.5/1729
4003_6014	Channel n Status register (PDB_CH0S)	32	R/W	0000_0000h	38.3.6/1730
4003_6018	Channel n Delay 0 register (PDB_CH0DLY0)	32	R/W	0000_0000h	38.3.7/1730
4003_601C	Channel n Delay 1 register (PDB_CH0DLY1)	32	R/W	0000_0000h	38.3.8/1731
4003_6038	Channel n Control register 1 (PDB_CH1C1)	32	R/W	0000_0000h	38.3.5/1729
4003_603C	Channel n Status register (PDB_CH1S)	32	R/W	0000_0000h	38.3.6/1730
4003_6040	Channel n Delay 0 register (PDB_CH1DLY0)	32	R/W	0000_0000h	38.3.7/1730
4003_6044	Channel n Delay 1 register (PDB_CH1DLY1)	32	R/W	0000_0000h	38.3.8/1731
4003_6150	DAC Interval Trigger n Control register (PDB_DACINTC0)	32	R/W	0000_0000h	38.3.9/1731
4003_6154	DAC Interval n register (PDB_DACINT0)	32	R/W	0000_0000h	38.3.10/ 1732
4003_6158	DAC Interval Trigger n Control register (PDB_DACINTC1)	32	R/W	0000_0000h	38.3.9/1731
4003_615C	DAC Interval n register (PDB_DACINT1)	32	R/W	0000_0000h	38.3.10/ 1732

38.3.1 Status and Control register (PDB_SC)

Address: 4003_6000h base + 0h offset = 4003_6000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												LDMOD		PDBEIE	0
W																SWTRIG
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMAEN	PRESCALER			TRGSEL				PDBEN	PDBIF	PDBIE	0	MULT		CONT	LDOK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDB_SC field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–18 LDMOD	Load Mode Select Selects the mode to load the MOD, IDLY, CH*n*DLY*m*, INT*x*, and PO*y*DLY registers, after 1 is written to LDOK. 00 The internal registers are loaded with the values from their buffers immediately after 1 is written to LDOK. 01 The internal registers are loaded with the values from their buffers when the PDB counter reaches the MOD register value after 1 is written to LDOK. 10 The internal registers are loaded with the values from their buffers when a trigger input event is detected after 1 is written to LDOK. 11 The internal registers are loaded with the values from their buffers when either the PDB counter reaches the MOD register value or a trigger input event is detected, after 1 is written to LDOK.
17 PDBEIE	PDB Sequence Error Interrupt Enable Enables the PDB sequence error interrupt. When this field is set, any of the PDB channel sequence error flags generates a PDB sequence error interrupt.

Table continues on the next page...

PDB_SC field descriptions (continued)

Field	Description
	0 PDB sequence error interrupt disabled. 1 PDB sequence error interrupt enabled.
16 SWTRIG	Software Trigger When PDB is enabled and the software trigger is selected as the trigger input source, writing 1 to this field resets and restarts the counter. Writing 0 to this field has no effect. Reading this field results 0.
15 DMAEN	DMA Enable When DMA is enabled, the PDBIF flag generates a DMA request instead of an interrupt. 0 DMA disabled. 1 DMA enabled.
14–12 PRESCALER	Prescaler Divider Select 000 Counting uses the peripheral clock divided by multiplication factor selected by MULT. 001 Counting uses the peripheral clock divided by twice of the multiplication factor selected by MULT. 010 Counting uses the peripheral clock divided by four times of the multiplication factor selected by MULT. 011 Counting uses the peripheral clock divided by eight times of the multiplication factor selected by MULT. 100 Counting uses the peripheral clock divided by 16 times of the multiplication factor selected by MULT. 101 Counting uses the peripheral clock divided by 32 times of the multiplication factor selected by MULT. 110 Counting uses the peripheral clock divided by 64 times of the multiplication factor selected by MULT. 111 Counting uses the peripheral clock divided by 128 times of the multiplication factor selected by MULT.
11–8 TRGSEL	Trigger Input Source Select Selects the trigger input source for the PDB. The trigger input source can be internal or external (EXTRG pin), or the software trigger. Please refer to Chip Configuration chapter for the actual PDB input trigger connections. 0000 Trigger-In 0 is selected. 0001 Trigger-In 1 is selected. 0010 Trigger-In 2 is selected. 0011 Trigger-In 3 is selected. 0100 Trigger-In 4 is selected. 0101 Trigger-In 5 is selected. 0110 Trigger-In 6 is selected. 0111 Trigger-In 7 is selected. 1000 Trigger-In 8 is selected. 1001 Trigger-In 9 is selected. 1010 Trigger-In 10 is selected. 1011 Trigger-In 11 is selected. 1100 Trigger-In 12 is selected. 1101 Trigger-In 13 is selected. 1110 Trigger-In 14 is selected. 1111 Software trigger is selected.

Table continues on the next page...

PDB_SC field descriptions (continued)

Field	Description
7 PDBEN	PDB Enable 0 PDB disabled. Counter is off. 1 PDB enabled.
6 PDBIF	PDB Interrupt Flag This field is set when the counter value is equal to the IDLY register. Writing zero clears this field.
5 PDBIE	PDB Interrupt Enable Enables the PDB interrupt. When this field is set and DMAEN is cleared, PDBIF generates a PDB interrupt. 0 PDB interrupt disabled. 1 PDB interrupt enabled.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–2 MULT	Multiplication Factor Select for Prescaler Selects the multiplication factor of the prescaler divider for the counter clock. 00 Multiplication factor is 1. 01 Multiplication factor is 10. 10 Multiplication factor is 20. 11 Multiplication factor is 40.
1 CONT	Continuous Mode Enable Enables the PDB operation in Continuous mode. 0 PDB operation in One-Shot mode 1 PDB operation in Continuous mode
0 LDOK	Load OK Writing 1 to this bit updates the internal registers of MOD, IDLY, CHnDLYm, DACINTx,and POyDLY with the values written to their buffers. The MOD, IDLY, CHnDLYm, DACINTx, and POyDLY will take effect according to the LDMOD. After 1 is written to the LDOK field, the values in the buffers of above registers are not effective and the buffers cannot be written until the values in buffers are loaded into their internal registers. LDOK can be written only when PDBEN is set or it can be written at the same time with PDBEN being written to 1. It is automatically cleared when the values in buffers are loaded into the internal registers or the PDBEN is cleared. Writing 0 to it has no effect.

38.3.2 Modulus register (PDB_MOD)

Address: 4003_6000h base + 4h offset = 4003_6004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

PDB_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 MOD	PDB Modulus Specifies the period of the counter. When the counter reaches this value, it will be reset back to zero. If the PDB is in Continuous mode, the count begins anew. Reading this field returns the value of the internal register that is effective for the current cycle of PDB.

38.3.3 Counter register (PDB_CNT)

Address: 4003_6000h base + 8h offset = 4003_6008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDB_CNT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 CNT	PDB Counter Contains the current value of the counter.

38.3.4 Interrupt Delay register (PDB_IDLY)

Address: 4003_6000h base + Ch offset = 4003_600Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																IDLY															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

PDB_IDLY field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 IDLY	PDB Interrupt Delay Specifies the delay value to schedule the PDB interrupt. It can be used to schedule an independent interrupt at some point in the PDB cycle. If enabled, a PDB interrupt is generated, when the counter is

Table continues on the next page...

PDB_IDLY field descriptions (continued)

Field	Description
	equal to the IDLY. Reading this field returns the value of internal register that is effective for the current cycle of the PDB.

38.3.5 Channel n Control register 1 (PDB_CH*n*C1)

Each PDB channel has one control register, CHnC1. The bits in this register control the functionality of each PDB channel operation.

Address: 4003_6000h base + 10h offset + (40d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								BB								TOS								EN							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDB_CH*n*C1 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 BB	PDB Channel Pre-Trigger Back-to-Back Operation Enable These bits enable the PDB ADC pre-trigger operation as back-to-back mode. Only lower M pre-trigger bits are implemented in this MCU. Back-to-back operation enables the ADC conversions complete to trigger the next PDB channel pre-trigger and trigger output, so that the ADC conversions can be triggered on next set of configuration and results registers. Application code must only enable the back-to-back operation of the PDB pre-triggers at the leading of the back-to-back connection chain. 0 PDB channel's corresponding pre-trigger back-to-back operation disabled. 1 PDB channel's corresponding pre-trigger back-to-back operation enabled.
15–8 TOS	PDB Channel Pre-Trigger Output Select Selects the PDB ADC pre-trigger outputs. Only lower M pre-trigger fields are implemented in this MCU. 0 PDB channel's corresponding pre-trigger is in bypassed mode. The pre-trigger asserts one peripheral clock cycle after a rising edge is detected on selected trigger input source or software trigger is selected and SWTRIG is written with 1. 1 PDB channel's corresponding pre-trigger asserts when the counter reaches the channel delay register and one peripheral clock cycle after a rising edge is detected on selected trigger input source or software trigger is selected and SETRIG is written with 1.
7–0 EN	PDB Channel Pre-Trigger Enable These bits enable the PDB ADC pre-trigger outputs. Only lower M pre-trigger bits are implemented in this MCU. 0 PDB channel's corresponding pre-trigger disabled. 1 PDB channel's corresponding pre-trigger enabled.

38.3.6 Channel n Status register (PDB_CH*n*S)

Address: 4003_6000h base + 14h offset + (40d × i), where i=0d to 1d

Bit	31–24	23–16	15–8	7–0
R	0	CF	0	ERR
W				
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

PDB_CH*n*S field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 CF	PDB Channel Flags The CF[*m*] bit is set when the PDB counter matches the CH*n*DLY*m*. Write 0 to clear these bits.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 ERR	PDB Channel Sequence Error Flags Only the lower M bits are implemented in this MCU. 0 Sequence error not detected on PDB channel's corresponding pre-trigger. 1 Sequence error detected on PDB channel's corresponding pre-trigger. ADCn block can be triggered for a conversion by one pre-trigger from PDB channel *n*. When one conversion, which is triggered by one of the pre-triggers from PDB channel *n*, is in progress, new trigger from PDB channel's corresponding pre-trigger m cannot be accepted by ADCn, and ERR[m] is set. Writing 0's to clear the sequence error flags.

38.3.7 Channel n Delay 0 register (PDB_CH*n*DLY0)

Address: 4003_6000h base + 18h offset + (40d × i), where i=0d to 1d

Bit	31–16	15–0
R	0	DLY
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PDB_CH*n*DLY0 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 DLY	PDB Channel Delay Specifies the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading this field returns the value of internal register that is effective for the current PDB cycle.

38.3.8 Channel n Delay 1 register (PDB_CH*n*DLY1)

Address: 4003_6000h base + 1Ch offset + (40d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																DLY															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDB_CH*n*DLY1 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

38.3.9 DAC Interval Trigger n Control register (PDB_DACINTC*n*)

Address: 4003_6000h base + 150h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														EXT	TOE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDB_DACINTC*n* field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 EXT	DAC External Trigger Input Enable Enables the external trigger for DAC interval counter. 0 DAC external trigger input disabled. DAC interval counter is reset and counting starts when a rising edge is detected on selected trigger input source or software trigger is selected and SWTRIG is written with 1. 1 DAC external trigger input enabled. DAC interval counter is bypassed and DAC external trigger input triggers the DAC interval trigger.
0 TOE	DAC Interval Trigger Enable

Table continues on the next page...

PDB_DACINTCn field descriptions (continued)

Field	Description
	This bit enables the DAC interval trigger. 0 DAC interval trigger disabled. 1 DAC interval trigger enabled.

38.3.10 DAC Interval n register (PDB_DACINT*n*)

Address: 4003_6000h base + 154h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																INT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDB_DACINTn field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 INT	DAC Interval Specifies the interval value for DAC interval trigger. DAC interval trigger triggers DAC[1:0] update when the DAC interval counter is equal to the DACINT. Reading this field returns the value of internal register that is effective for the current PDB cycle.

38.4 Functional description

38.4.1 PDB pre-trigger and trigger outputs

The PDB contains a counter whose output is compared against several different digital values. If the PDB is enabled, a trigger input event will reset the counter and make it start to count. A trigger input event is defined as a rising edge being detected on selected trigger input source or software trigger being selected and SC[SWTRIG] is written with 1. For each channel, delay *m* determines the time between assertion of the trigger input event to the point at which changes in the pre-trigger *m* output signal is initiated. The time is defined as:

- Trigger input event to pre-trigger *m* = (prescaler X multiplication factor X delay *m*) + 2 peripheral clock cycles
- Add one additional peripheral clock cycle to determine the time at which the channel trigger output change.

Each channel is associated with one ADC block. PDB channel n pre-trigger outputs 0 to *M* and trigger output is connected to ADC hardware trigger select and hardware trigger inputs. The pre-triggers are used to precondition the ADC block prior to the actual trigger. The ADC contains *M* sets of configuration and result registers, allowing it to operate in a ping-pong fashion, alternating conversions between *M* different analog sources. The pre-trigger outputs are used to specify which signal will be sampled next. When pre-trigger *m* is asserted, the ADC conversion is triggered with set *m* of the configuration and result registers.

The waveforms shown in the following diagram illustrate the pre-trigger and trigger outputs of PDB channel *n*. The delays can be independently set via the CH*n*DLY*m* registers. And the pre-triggers can be enabled or disabled in CH*n*C1[EN[*m*]].

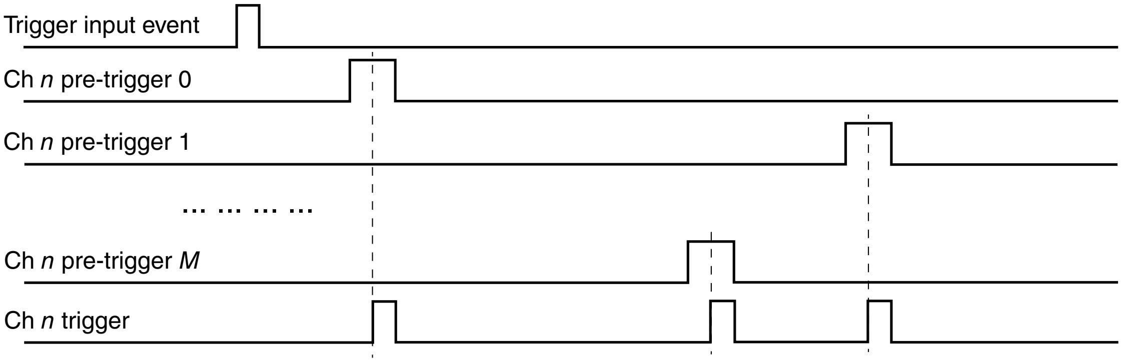


Figure 38-24. Pre-trigger and trigger outputs

The delay in CH*n*DLY*m* register can be optionally bypassed, if CH*n*C1[TOS[*m*]] is cleared. In this case, when the trigger input event occurs, the pre-trigger *m* is asserted after two peripheral clock cycles.

The PDB can be configured in back-to-back operation. Back-to-back operation enables the ADC conversions complete to trigger the next PDB channel pre-trigger and trigger outputs, so that the ADC conversions can be triggered on the next set of configuration and results registers. When back-to-back is enabled by setting CH*n*C1[BB[*m*]], the delay *m* is ignored and the pre-trigger *m* is asserted two peripheral cycles after the acknowledgment *m* is received. The acknowledgment connections in this MCU is described in Back-to-back acknowledgment connections.

When an ADC conversion, which is triggered by one of the pre-triggers from PDB channel *n*, is in progress and ADC*n*SC1[COCO] is not set, a new trigger from PDB channel *n* pre-trigger *m* cannot be accepted by ADC*n*. Therefore every time when one PDB channel *n* pre-trigger and trigger output starts an ADC conversion, an internal lock associated with the corresponding pre-trigger is activated. This lock becomes inactive when:

- the corresponding ADC*n*SC1[COCO] is set
- the corresponding PDB pre-trigger is disabled, or
- the PDB is disabled

The channel *n* trigger output is suppressed when any of the locks of the pre-triggers in channel *n* is active. If a new pre-trigger *m* asserts when there is active lock in the PDB channel *n*, a register flag bit, CH*n*S[ERR[*m*]], associated with the pre-trigger *m* is set. If SC[PDBEIE] is set, the sequence error interrupt is generated. Sequence error typically happens because the delay *m* is set too short and the pre-trigger *m* asserts before the previously triggered ADC conversion is completed.

When the PDB counter reaches the value set in IDLY register, the SC[PDBIF] flag is set. A PDB interrupt can be generated if SC[PDBIE] is set and SC[DMAEN] is cleared. If SC[DMAEN] is set, PDB requests a DMA transfer when SC[PDBIF] is set.

The modulus value in the MOD register is used to reset the counter back to zero at the end of the count. If SC[CONT] is set, the counter will then resume a new count. Otherwise, the counter operation will cease until the next trigger input event occurs.

38.4.2 PDB trigger input source selection

The PDB has up to 15 trigger input sources, namely Trigger-In 14. They are connected to on-chip or off-chip event sources. The PDB can be triggered by software through SC[SWTRIG]. SC[TRIGSEL] selects the active trigger input source or software trigger.

For the trigger input sources implemented in this MCU, see chip configuration information.

38.4.3 DAC interval trigger outputs

PDB can generate the interval triggers for DACs to update their outputs periodically. DAC interval counter *x* is reset and started when a trigger input event occurs if DACINTC*x*[EXT] is cleared. When the interval counter *x* is equal to the value set in DACINT*x* register, the DAC interval trigger *x* output generates a pulse of one peripheral clock cycle width to update the DAC*x*. If DACINTC*x*[EXT] is set, the DAC interval

counter is bypassed and the interval trigger output *x* generates a pulse following the detection of a rising edge on the DAC external trigger input. The counter and interval trigger can be disabled by clearing the DACINTC*x*[TOE].

DAC interval counters are also reset when the PDB counter reaches the MOD register value; therefore, when the PDB counter rolls over to zero, the DAC interval counters starts anew.

The DAC interval trigger pulse and the ADC pre-trigger/trigger pulses together allow precise timing of DAC updates and ADC measurements. This is outlined in the typical use case described in the following diagram.

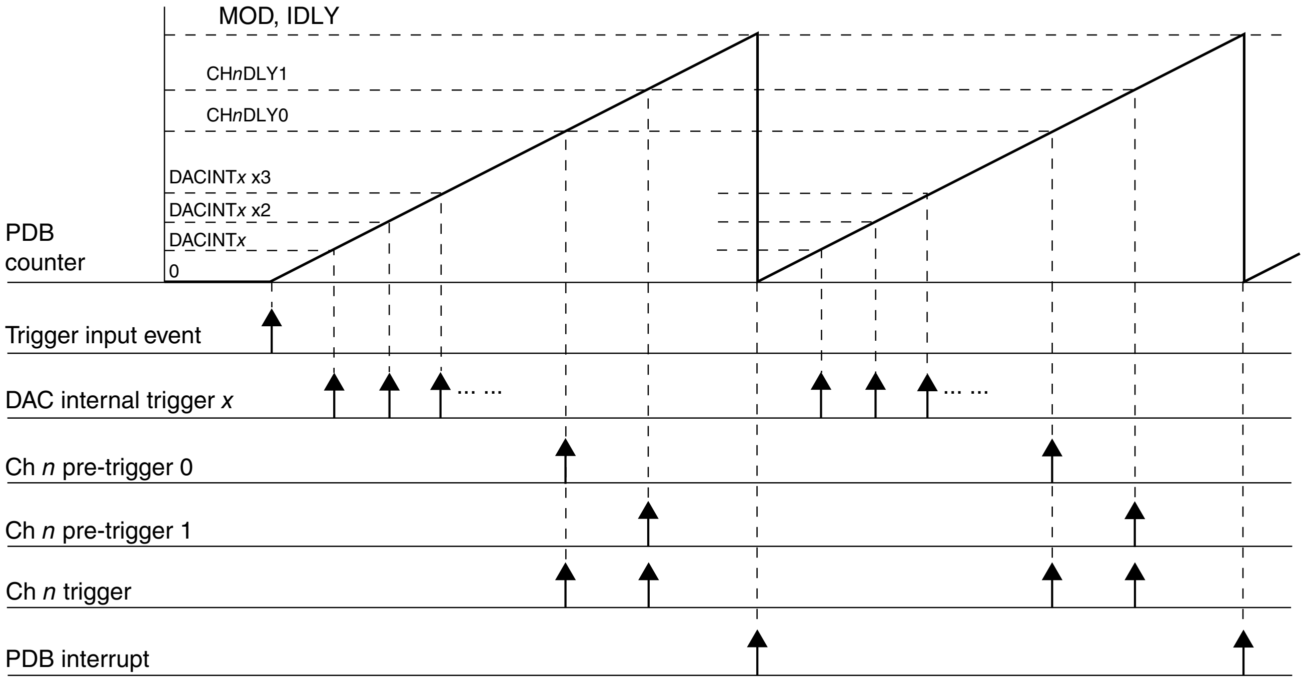


Figure 38-25. PDB ADC triggers and DAC interval triggers use case

NOTE

Because the DAC interval counters share the prescaler with PDB counter, PDB must be enabled if the DAC interval trigger outputs are used in the applications.

38.4.4 Updating the delay registers

The following registers control the timing of the PDB operation; and in some of the applications, they may need to become effective at the same time.

- PDB Modulus register (MOD)

- PDB Interrupt Delay register (IDLY)
- PDB Channel *n* Delay *m* register (CH*n*DLY*m*)
- DAC Interval *x* register (DACINT*x*)
- PDB Pulse-Out *y* Delay register (PO*y*DLY)

The internal registers of them are buffered and any values written to them are written first to their buffers. The circumstances that cause their internal registers to be updated with the values from the buffers are summarized as shown in the table below.

Table 38-25. Circumstances of update to the delay registers

SC[LDMOD]	Update to the delay registers
00	The internal registers are loaded with the values from their buffers immediately after 1 is written to SC[LDOK].
01	The PDB counter reaches the MOD register value after 1 is written to SC[LDOK].
10	A trigger input event is detected after 1 is written to SC[LDOK].
11	Either the PDB counter reaches the MOD register value, or a trigger input event is detected, after 1 is written to SC[LDOK].

After 1 is written to SC[LDOK], the buffers cannot be written until the values in buffers are loaded into their internal registers. SC[LDOK] is self-cleared when the internal registers are loaded, so the application code can read it to determine the updates to the internal registers.

The following diagrams show the cases of the internal registers being updated with SC[LDMOD] is 00 and x1.

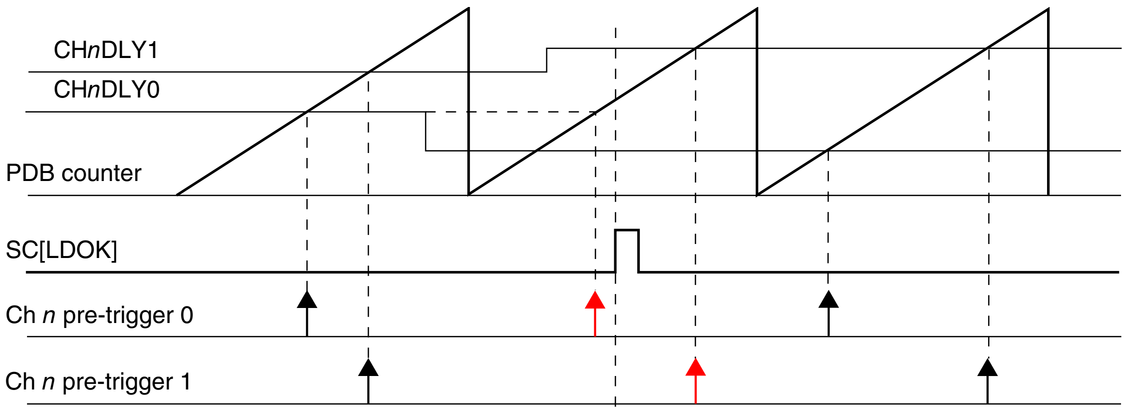


Figure 38-26. Registers update with SC[LDMOD] = 00

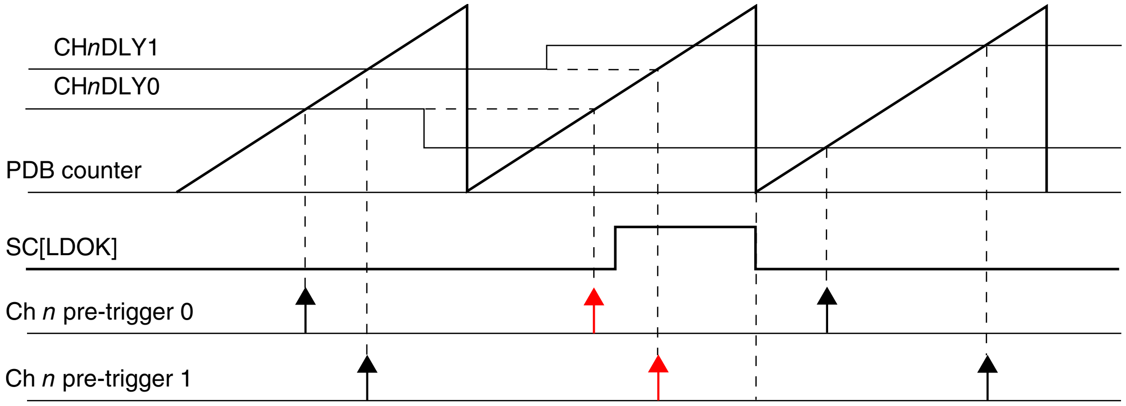


Figure 38-27. Registers update with SC[LDMOD] = x1

38.4.5 Interrupts

PDB can generate two interrupts: PDB interrupt and PDB sequence error interrupt. The following table summarizes the interrupts.

Table 38-26. PDB interrupt summary

Interrupt	Flags	Enable bit
PDB Interrupt	SC[PDBIF]	SC[PDBIE] = 1 and SC[DMAEN] = 0
PDB Sequence Error Interrupt	CH*n*S[ERR*m*]	SC[PDBEIE] = 1

38.4.6 DMA

If SC[DMAEN] is set, PDB can generate a DMA transfer request when SC[PDBIF] is set. When DMA is enabled, the PDB interrupt is not issued.

38.5 Application information

38.5.1 Impact of using the prescaler and multiplication factor on timing resolution

Use of prescaler and multiplication factor greater than 1 limits the count/delay accuracy in terms of peripheral clock cycles (to the modulus of the prescaler X multiplication factor). If the multiplication factor is set to 1 and the prescaler is set to 2 then the only

values of total peripheral clocks that can be detected are even values; if prescaler is set to 4 then the only values of total peripheral clocks that can be decoded as detected are mod(4) and so forth. If the applications need a really long delay value and use a prescaler set to 128, then the resolution would be limited to 128 peripheral clock cycles.

Therefore, use the lowest possible prescaler and multiplication factor for a given application.

Chapter 39
FlexTimer Module (FTM)

39.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The FlexTimer module (FTM) is a two-to-eight channel timer that supports input capture, output compare, and the generation of PWM signals to control electric motor and power management applications. The FTM time reference is a 16-bit counter that can be used as an unsigned or signed counter.

39.1.1 FlexTimer philosophy

The FlexTimer is built upon a simple timer, the HCS08 Timer PWM Module – TPM, used for many years on Freescale's 8-bit microcontrollers. The FlexTimer extends the functionality to meet the demands of motor control, digital lighting solutions, and power conversion, while providing low cost and backwards compatibility with the TPM module.

Several key enhancements are made:

- Signed up counter
- Deadtime insertion hardware
- Fault control inputs
- Enhanced triggering functionality
- Initialization and polarity control

All of the features common with the TPM have fully backwards compatible register assignments. The FlexTimer can also use code on the same core platform without change to perform the same functions.

Motor control and power conversion features have been added through a dedicated set of registers and defaults turn off all new features. The new features, such as hardware deadtime insertion, polarity, fault control, and output forcing and masking, greatly reduce loading on the execution software and are usually each controlled by a group of registers.

FlexTimer input triggers can be from comparators, ADC, or other submodules to initiate timer functions automatically. These triggers can be linked in a variety of ways during integration of the sub modules so please note the options available for used FlexTimer configuration.

Several FlexTimers may be synchronized to provide a larger timer with their counters incrementing in unison, assuming the initialization, the input clocks, the initial and final counting values are the same in each FlexTimer.

All main user access registers are buffered to ease the load on the executing software. A number of trigger options exist to determine which registers are updated with this user defined data.

39.1.2 Features

The FTM features include:

- FTM source clock is selectable
    - Source clock can be the system clock, the fixed frequency clock, or an external clock
    - Fixed frequency clock is an additional clock input to allow the selection of an on chip clock source other than the system clock
    - Selecting external clock connects FTM clock to a chip level input pin therefore allowing to synchronize the FTM counter with an off chip clock source
- Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128
- 16-bit counter
    - It can be a free-running counter or a counter with initial and final value
    - The counting can be up or up-down
- Each channel can be configured for input capture, output compare, or edge-aligned PWM mode
- In Input Capture mode:

  - The capture can occur on rising edges, falling edges or both edges
  - An input filter can be selected for some channels
- In Output Compare mode the output signal can be set, cleared, or toggled on match
- All channels can be configured for center-aligned PWM mode
- Each pair of channels can be combined to generate a PWM signal with independent control of both edges of PWM signal
- The FTM channels can operate as pairs with equal outputs, pairs with complementary outputs, or independent channels with independent outputs
- The deadtime insertion is available for each complementary pair
- Generation of match triggers
- Software control of PWM outputs
- Up to 4 fault inputs for global fault control
- The polarity of each channel is configurable
- The generation of an interrupt per channel
- The generation of an interrupt when the counter overflows
- The generation of an interrupt when the fault condition is detected
- Synchronized loading of write buffered FTM registers
- Write protection for critical registers
- Backwards compatible with TPM
- Testing of input captures for a stuck at zero and one conditions
- Dual edge capture for pulse and period width measurement
- Quadrature decoder with input filters, relative position counting, and interrupt on position count or capture of position count on external event

39.1.3 Modes of operation

When the MCU is in an active BDM mode, the FTM temporarily suspends all counting until the MCU returns to normal user operating mode. During Stop mode, all FTM input clocks are stopped, so the FTM is effectively disabled until clocks resume. During Wait mode, the FTM continues to operate normally. If the FTM does not need to produce a

real time reference or provide the interrupt sources needed to wake the MCU from Wait mode, the power can then be saved by disabling FTM functions before entering Wait mode.

39.1.4 Block diagram

The FTM uses one input/output (I/O) pin per channel, CHn (FTM channel (n)) where n is the channel number (0–7).

The following figure shows the FTM structure. The central component of the FTM is the 16-bit counter with programmable initial and final values and its counting can be up or up-down.

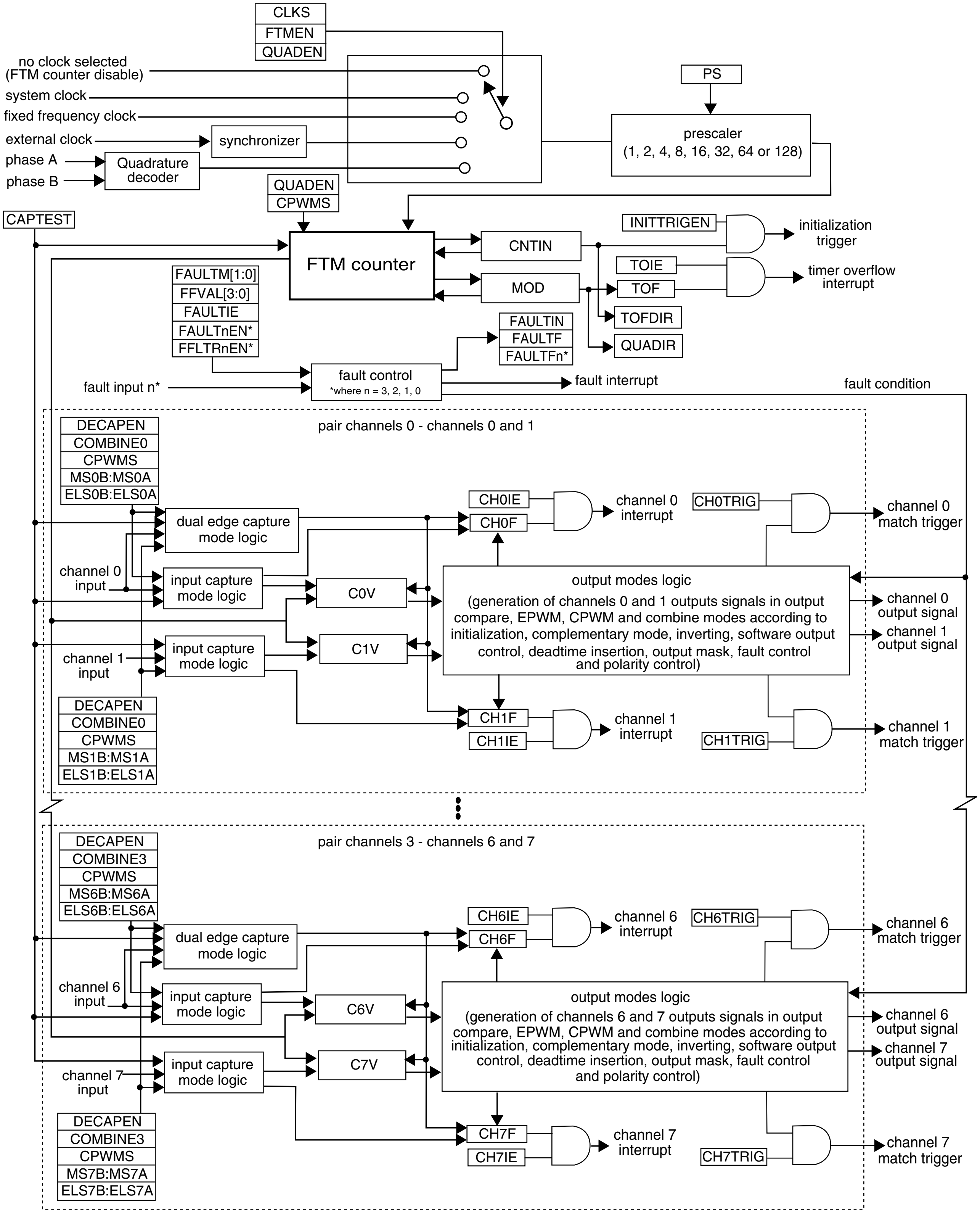


Figure 39-1. FTM block diagram

39.2 FTM signal descriptions

Table 39-1 shows the user-accessible signals for the FTM.

Table 39-1. FTM signal descriptions

Signal	Description	I/O	Function
EXTCLK	External clock. FTM external clock can be selected to drive the FTM counter.	I	The external clock input signal is used as the FTM counter clock if selected by CLKS[1:0] bits in the SC register. This clock signal must not exceed 1/4 of system clock frequency. The FTM counter prescaler selection and settings are also used when an external clock is selected.
CHn	FTM channel (n), where n can be 7-0	I/O	Each FTM channel can be configured to operate either as input or output. The direction associated with each channel, input or output, is selected according to the mode assigned for that channel.
FAULTj	Fault input (j), where j can be 3-0	I	The fault input signals are used to control the CHn channel output state. If a fault is detected, the FAULTj signal is asserted and the channel output is put in a safe state. The behavior of the fault logic is defined by the FAULTM[1:0] control bits in the MODE register and FAULTEN bit in the COMBINEm register. Note that each FAULTj input may affect all channels selectively since FAULTM[1:0] and FAULTEN control bits are defined for each pair of channels. Because there are several FAULTj inputs, maximum of 4 for the FTM module, each one of these inputs is activated by the FAULTjEN bit in the FLTCTRL register.
PHA	Quadrature decoder phase A input. Input pin associated with quadrature decoder phase A.	I	The quadrature decoder phase A input is used as the Quadrature Decoder mode is selected. The phase A input signal is one of the signals that control the FTM counter increment or decrement in the Quadrature Decoder mode.
PHB	Quadrature decoder phase B input. Input pin associated with quadrature decoder phase B.	I	The quadrature decoder phase B input is used as the Quadrature Decoder mode is selected. The phase B input signal is one of the signals that control the FTM counter increment or decrement in the Quadrature Decoder mode.

39.3 Memory map and register definition

39.3.1 Memory map

This section presents a high-level summary of the FTM registers and how they are mapped.

The registers and bits of an unavailable function in the FTM remain in the memory map and in the reset value, but they have no active function.

Note

Do not write in the region from the CNTIN register through the PWMLOAD register when FTMEN = 0.

39.3.2 Register descriptions

Accesses to reserved addresses result in transfer errors. Registers for absent channels are considered reserved.

FTM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8000	Status And Control (FTM0_SC)	32	R/W	0000_0000h	39.3.3/1751
4003_8004	Counter (FTM0_CNT)	32	R/W	0000_0000h	39.3.4/1752
4003_8008	Modulo (FTM0_MOD)	32	R/W	0000_0000h	39.3.5/1753
4003_800C	Channel (n) Status And Control (FTM0_C0SC)	32	R/W	0000_0000h	39.3.6/1754
4003_8010	Channel (n) Value (FTM0_C0V)	32	R/W	0000_0000h	39.3.7/1756
4003_8014	Channel (n) Status And Control (FTM0_C1SC)	32	R/W	0000_0000h	39.3.6/1754
4003_8018	Channel (n) Value (FTM0_C1V)	32	R/W	0000_0000h	39.3.7/1756
4003_801C	Channel (n) Status And Control (FTM0_C2SC)	32	R/W	0000_0000h	39.3.6/1754
4003_8020	Channel (n) Value (FTM0_C2V)	32	R/W	0000_0000h	39.3.7/1756
4003_8024	Channel (n) Status And Control (FTM0_C3SC)	32	R/W	0000_0000h	39.3.6/1754
4003_8028	Channel (n) Value (FTM0_C3V)	32	R/W	0000_0000h	39.3.7/1756
4003_802C	Channel (n) Status And Control (FTM0_C4SC)	32	R/W	0000_0000h	39.3.6/1754
4003_8030	Channel (n) Value (FTM0_C4V)	32	R/W	0000_0000h	39.3.7/1756
4003_8034	Channel (n) Status And Control (FTM0_C5SC)	32	R/W	0000_0000h	39.3.6/1754
4003_8038	Channel (n) Value (FTM0_C5V)	32	R/W	0000_0000h	39.3.7/1756
4003_803C	Channel (n) Status And Control (FTM0_C6SC)	32	R/W	0000_0000h	39.3.6/1754
4003_8040	Channel (n) Value (FTM0_C6V)	32	R/W	0000_0000h	39.3.7/1756
4003_8044	Channel (n) Status And Control (FTM0_C7SC)	32	R/W	0000_0000h	39.3.6/1754
4003_8048	Channel (n) Value (FTM0_C7V)	32	R/W	0000_0000h	39.3.7/1756
4003_804C	Counter Initial Value (FTM0_CNTIN)	32	R/W	0000_0000h	39.3.8/1757
4003_8050	Capture And Compare Status (FTM0_STATUS)	32	R/W	0000_0000h	39.3.9/1757
4003_8054	Features Mode Selection (FTM0_MODE)	32	R/W	0000_0004h	39.3.10/ 1759
4003_8058	Synchronization (FTM0_SYNC)	32	R/W	0000_0000h	39.3.11/ 1761
4003_805C	Initial State For Channels Output (FTM0_OUTINIT)	32	R/W	0000_0000h	39.3.12/ 1764
4003_8060	Output Mask (FTM0_OUTMASK)	32	R/W	0000_0000h	39.3.13/ 1765

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8064	Function For Linked Channels (FTM0_COMBINE)	32	R/W	0000_0000h	39.3.14/ 1767
4003_8068	Deadtime Insertion Control (FTM0_DEADTIME)	32	R/W	0000_0000h	39.3.15/ 1772
4003_806C	FTM External Trigger (FTM0_EXTTRIG)	32	R/W	0000_0000h	39.3.16/ 1773
4003_8070	Channels Polarity (FTM0_POL)	32	R/W	0000_0000h	39.3.17/ 1775
4003_8074	Fault Mode Status (FTM0_FMS)	32	R/W	0000_0000h	39.3.18/ 1777
4003_8078	Input Capture Filter Control (FTM0_FILTER)	32	R/W	0000_0000h	39.3.19/ 1779
4003_807C	Fault Control (FTM0_FLTCTRL)	32	R/W	0000_0000h	39.3.20/ 1780
4003_8080	Quadrature Decoder Control And Status (FTM0_QDCTRL)	32	R/W	0000_0000h	39.3.21/ 1782
4003_8084	Configuration (FTM0_CONF)	32	R/W	0000_0000h	39.3.22/ 1784
4003_8088	FTM Fault Input Polarity (FTM0_FLTPOL)	32	R/W	0000_0000h	39.3.23/ 1785
4003_808C	Synchronization Configuration (FTM0_SYNCONF)	32	R/W	0000_0000h	39.3.24/ 1787
4003_8090	FTM Inverting Control (FTM0_INVCTRL)	32	R/W	0000_0000h	39.3.25/ 1789
4003_8094	FTM Software Output Control (FTM0_SWOCTRL)	32	R/W	0000_0000h	39.3.26/ 1790
4003_8098	FTM PWM Load (FTM0_PWMLOAD)	32	R/W	0000_0000h	39.3.27/ 1792
4003_9000	Status And Control (FTM1_SC)	32	R/W	0000_0000h	39.3.3/1751
4003_9004	Counter (FTM1_CNT)	32	R/W	0000_0000h	39.3.4/1752
4003_9008	Modulo (FTM1_MOD)	32	R/W	0000_0000h	39.3.5/1753
4003_900C	Channel (n) Status And Control (FTM1_C0SC)	32	R/W	0000_0000h	39.3.6/1754
4003_9010	Channel (n) Value (FTM1_C0V)	32	R/W	0000_0000h	39.3.7/1756
4003_9014	Channel (n) Status And Control (FTM1_C1SC)	32	R/W	0000_0000h	39.3.6/1754
4003_9018	Channel (n) Value (FTM1_C1V)	32	R/W	0000_0000h	39.3.7/1756
4003_901C	Channel (n) Status And Control (FTM1_C2SC)	32	R/W	0000_0000h	39.3.6/1754
4003_9020	Channel (n) Value (FTM1_C2V)	32	R/W	0000_0000h	39.3.7/1756
4003_9024	Channel (n) Status And Control (FTM1_C3SC)	32	R/W	0000_0000h	39.3.6/1754
4003_9028	Channel (n) Value (FTM1_C3V)	32	R/W	0000_0000h	39.3.7/1756
4003_902C	Channel (n) Status And Control (FTM1_C4SC)	32	R/W	0000_0000h	39.3.6/1754
4003_9030	Channel (n) Value (FTM1_C4V)	32	R/W	0000_0000h	39.3.7/1756
4003_9034	Channel (n) Status And Control (FTM1_C5SC)	32	R/W	0000_0000h	39.3.6/1754

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_9038	Channel (n) Value (FTM1_C5V)	32	R/W	0000_0000h	39.3.7/1756
4003_903C	Channel (n) Status And Control (FTM1_C6SC)	32	R/W	0000_0000h	39.3.6/1754
4003_9040	Channel (n) Value (FTM1_C6V)	32	R/W	0000_0000h	39.3.7/1756
4003_9044	Channel (n) Status And Control (FTM1_C7SC)	32	R/W	0000_0000h	39.3.6/1754
4003_9048	Channel (n) Value (FTM1_C7V)	32	R/W	0000_0000h	39.3.7/1756
4003_904C	Counter Initial Value (FTM1_CNTIN)	32	R/W	0000_0000h	39.3.8/1757
4003_9050	Capture And Compare Status (FTM1_STATUS)	32	R/W	0000_0000h	39.3.9/1757
4003_9054	Features Mode Selection (FTM1_MODE)	32	R/W	0000_0004h	39.3.10/ 1759
4003_9058	Synchronization (FTM1_SYNC)	32	R/W	0000_0000h	39.3.11/ 1761
4003_905C	Initial State For Channels Output (FTM1_OUTINIT)	32	R/W	0000_0000h	39.3.12/ 1764
4003_9060	Output Mask (FTM1_OUTMASK)	32	R/W	0000_0000h	39.3.13/ 1765
4003_9064	Function For Linked Channels (FTM1_COMBINE)	32	R/W	0000_0000h	39.3.14/ 1767
4003_9068	Deadtime Insertion Control (FTM1_DEADTIME)	32	R/W	0000_0000h	39.3.15/ 1772
4003_906C	FTM External Trigger (FTM1_EXTTRIG)	32	R/W	0000_0000h	39.3.16/ 1773
4003_9070	Channels Polarity (FTM1_POL)	32	R/W	0000_0000h	39.3.17/ 1775
4003_9074	Fault Mode Status (FTM1_FMS)	32	R/W	0000_0000h	39.3.18/ 1777
4003_9078	Input Capture Filter Control (FTM1_FILTER)	32	R/W	0000_0000h	39.3.19/ 1779
4003_907C	Fault Control (FTM1_FLTCTRL)	32	R/W	0000_0000h	39.3.20/ 1780
4003_9080	Quadrature Decoder Control And Status (FTM1_QDCTRL)	32	R/W	0000_0000h	39.3.21/ 1782
4003_9084	Configuration (FTM1_CONF)	32	R/W	0000_0000h	39.3.22/ 1784
4003_9088	FTM Fault Input Polarity (FTM1_FLTPOL)	32	R/W	0000_0000h	39.3.23/ 1785
4003_908C	Synchronization Configuration (FTM1_SYNCONF)	32	R/W	0000_0000h	39.3.24/ 1787
4003_9090	FTM Inverting Control (FTM1_INVCTRL)	32	R/W	0000_0000h	39.3.25/ 1789
4003_9094	FTM Software Output Control (FTM1_SWOCTRL)	32	R/W	0000_0000h	39.3.26/ 1790
4003_9098	FTM PWM Load (FTM1_PWMLOAD)	32	R/W	0000_0000h	39.3.27/ 1792

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_8000	Status And Control (FTM2_SC)	32	R/W	0000_0000h	39.3.3/1751
400B_8004	Counter (FTM2_CNT)	32	R/W	0000_0000h	39.3.4/1752
400B_8008	Modulo (FTM2_MOD)	32	R/W	0000_0000h	39.3.5/1753
400B_800C	Channel (n) Status And Control (FTM2_C0SC)	32	R/W	0000_0000h	39.3.6/1754
400B_8010	Channel (n) Value (FTM2_C0V)	32	R/W	0000_0000h	39.3.7/1756
400B_8014	Channel (n) Status And Control (FTM2_C1SC)	32	R/W	0000_0000h	39.3.6/1754
400B_8018	Channel (n) Value (FTM2_C1V)	32	R/W	0000_0000h	39.3.7/1756
400B_801C	Channel (n) Status And Control (FTM2_C2SC)	32	R/W	0000_0000h	39.3.6/1754
400B_8020	Channel (n) Value (FTM2_C2V)	32	R/W	0000_0000h	39.3.7/1756
400B_8024	Channel (n) Status And Control (FTM2_C3SC)	32	R/W	0000_0000h	39.3.6/1754
400B_8028	Channel (n) Value (FTM2_C3V)	32	R/W	0000_0000h	39.3.7/1756
400B_802C	Channel (n) Status And Control (FTM2_C4SC)	32	R/W	0000_0000h	39.3.6/1754
400B_8030	Channel (n) Value (FTM2_C4V)	32	R/W	0000_0000h	39.3.7/1756
400B_8034	Channel (n) Status And Control (FTM2_C5SC)	32	R/W	0000_0000h	39.3.6/1754
400B_8038	Channel (n) Value (FTM2_C5V)	32	R/W	0000_0000h	39.3.7/1756
400B_803C	Channel (n) Status And Control (FTM2_C6SC)	32	R/W	0000_0000h	39.3.6/1754
400B_8040	Channel (n) Value (FTM2_C6V)	32	R/W	0000_0000h	39.3.7/1756
400B_8044	Channel (n) Status And Control (FTM2_C7SC)	32	R/W	0000_0000h	39.3.6/1754
400B_8048	Channel (n) Value (FTM2_C7V)	32	R/W	0000_0000h	39.3.7/1756
400B_804C	Counter Initial Value (FTM2_CNTIN)	32	R/W	0000_0000h	39.3.8/1757
400B_8050	Capture And Compare Status (FTM2_STATUS)	32	R/W	0000_0000h	39.3.9/1757
400B_8054	Features Mode Selection (FTM2_MODE)	32	R/W	0000_0004h	39.3.10/ 1759
400B_8058	Synchronization (FTM2_SYNC)	32	R/W	0000_0000h	39.3.11/ 1761
400B_805C	Initial State For Channels Output (FTM2_OUTINIT)	32	R/W	0000_0000h	39.3.12/ 1764
400B_8060	Output Mask (FTM2_OUTMASK)	32	R/W	0000_0000h	39.3.13/ 1765
400B_8064	Function For Linked Channels (FTM2_COMBINE)	32	R/W	0000_0000h	39.3.14/ 1767
400B_8068	Deadtime Insertion Control (FTM2_DEADTIME)	32	R/W	0000_0000h	39.3.15/ 1772
400B_806C	FTM External Trigger (FTM2_EXTTRIG)	32	R/W	0000_0000h	39.3.16/ 1773
400B_8070	Channels Polarity (FTM2_POL)	32	R/W	0000_0000h	39.3.17/ 1775
400B_8074	Fault Mode Status (FTM2_FMS)	32	R/W	0000_0000h	39.3.18/ 1777
400B_8078	Input Capture Filter Control (FTM2_FILTER)	32	R/W	0000_0000h	39.3.19/ 1779

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_807C	Fault Control (FTM2_FLTCTRL)	32	R/W	0000_0000h	39.3.20/ 1780
400B_8080	Quadrature Decoder Control And Status (FTM2_QDCTRL)	32	R/W	0000_0000h	39.3.21/ 1782
400B_8084	Configuration (FTM2_CONF)	32	R/W	0000_0000h	39.3.22/ 1784
400B_8088	FTM Fault Input Polarity (FTM2_FLTPOL)	32	R/W	0000_0000h	39.3.23/ 1785
400B_808C	Synchronization Configuration (FTM2_SYNCONF)	32	R/W	0000_0000h	39.3.24/ 1787
400B_8090	FTM Inverting Control (FTM2_INVCTRL)	32	R/W	0000_0000h	39.3.25/ 1789
400B_8094	FTM Software Output Control (FTM2_SWOCTRL)	32	R/W	0000_0000h	39.3.26/ 1790
400B_8098	FTM PWM Load (FTM2_PWMLOAD)	32	R/W	0000_0000h	39.3.27/ 1792
400B_9000	Status And Control (FTM3_SC)	32	R/W	0000_0000h	39.3.3/1751
400B_9004	Counter (FTM3_CNT)	32	R/W	0000_0000h	39.3.4/1752
400B_9008	Modulo (FTM3_MOD)	32	R/W	0000_0000h	39.3.5/1753
400B_900C	Channel (n) Status And Control (FTM3_C0SC)	32	R/W	0000_0000h	39.3.6/1754
400B_9010	Channel (n) Value (FTM3_C0V)	32	R/W	0000_0000h	39.3.7/1756
400B_9014	Channel (n) Status And Control (FTM3_C1SC)	32	R/W	0000_0000h	39.3.6/1754
400B_9018	Channel (n) Value (FTM3_C1V)	32	R/W	0000_0000h	39.3.7/1756
400B_901C	Channel (n) Status And Control (FTM3_C2SC)	32	R/W	0000_0000h	39.3.6/1754
400B_9020	Channel (n) Value (FTM3_C2V)	32	R/W	0000_0000h	39.3.7/1756
400B_9024	Channel (n) Status And Control (FTM3_C3SC)	32	R/W	0000_0000h	39.3.6/1754
400B_9028	Channel (n) Value (FTM3_C3V)	32	R/W	0000_0000h	39.3.7/1756
400B_902C	Channel (n) Status And Control (FTM3_C4SC)	32	R/W	0000_0000h	39.3.6/1754
400B_9030	Channel (n) Value (FTM3_C4V)	32	R/W	0000_0000h	39.3.7/1756
400B_9034	Channel (n) Status And Control (FTM3_C5SC)	32	R/W	0000_0000h	39.3.6/1754
400B_9038	Channel (n) Value (FTM3_C5V)	32	R/W	0000_0000h	39.3.7/1756
400B_903C	Channel (n) Status And Control (FTM3_C6SC)	32	R/W	0000_0000h	39.3.6/1754
400B_9040	Channel (n) Value (FTM3_C6V)	32	R/W	0000_0000h	39.3.7/1756
400B_9044	Channel (n) Status And Control (FTM3_C7SC)	32	R/W	0000_0000h	39.3.6/1754
400B_9048	Channel (n) Value (FTM3_C7V)	32	R/W	0000_0000h	39.3.7/1756
400B_904C	Counter Initial Value (FTM3_CNTIN)	32	R/W	0000_0000h	39.3.8/1757
400B_9050	Capture And Compare Status (FTM3_STATUS)	32	R/W	0000_0000h	39.3.9/1757
400B_9054	Features Mode Selection (FTM3_MODE)	32	R/W	0000_0004h	39.3.10/ 1759
400B_9058	Synchronization (FTM3_SYNC)	32	R/W	0000_0000h	39.3.11/ 1761

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_905C	Initial State For Channels Output (FTM3_OUTINIT)	32	R/W	0000_0000h	39.3.12/ 1764
400B_9060	Output Mask (FTM3_OUTMASK)	32	R/W	0000_0000h	39.3.13/ 1765
400B_9064	Function For Linked Channels (FTM3_COMBINE)	32	R/W	0000_0000h	39.3.14/ 1767
400B_9068	Deadtime Insertion Control (FTM3_DEADTIME)	32	R/W	0000_0000h	39.3.15/ 1772
400B_906C	FTM External Trigger (FTM3_EXTTRIG)	32	R/W	0000_0000h	39.3.16/ 1773
400B_9070	Channels Polarity (FTM3_POL)	32	R/W	0000_0000h	39.3.17/ 1775
400B_9074	Fault Mode Status (FTM3_FMS)	32	R/W	0000_0000h	39.3.18/ 1777
400B_9078	Input Capture Filter Control (FTM3_FILTER)	32	R/W	0000_0000h	39.3.19/ 1779
400B_907C	Fault Control (FTM3_FLTCTRL)	32	R/W	0000_0000h	39.3.20/ 1780
400B_9080	Quadrature Decoder Control And Status (FTM3_QDCTRL)	32	R/W	0000_0000h	39.3.21/ 1782
400B_9084	Configuration (FTM3_CONF)	32	R/W	0000_0000h	39.3.22/ 1784
400B_9088	FTM Fault Input Polarity (FTM3_FLTPOL)	32	R/W	0000_0000h	39.3.23/ 1785
400B_908C	Synchronization Configuration (FTM3_SYNCONF)	32	R/W	0000_0000h	39.3.24/ 1787
400B_9090	FTM Inverting Control (FTM3_INVCTRL)	32	R/W	0000_0000h	39.3.25/ 1789
400B_9094	FTM Software Output Control (FTM3_SWOCTRL)	32	R/W	0000_0000h	39.3.26/ 1790
400B_9098	FTM PWM Load (FTM3_PWMLOAD)	32	R/W	0000_0000h	39.3.27/ 1792

39.3.3 Status And Control (FTMx_SC)

SC contains the overflow status flag and control bits used to configure the interrupt enable, FTM configuration, clock source, and prescaler factor. These controls relate to all channels within this module.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TOF	TOIE	CPWMS	CLKS		PS		
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TOF	Timer Overflow Flag Set by hardware when the FTM counter passes the value in the MOD register. The TOF bit is cleared by reading the SC register while TOF is set and then writing a 0 to TOF bit. Writing a 1 to TOF has no effect. If another FTM overflow occurs between the read and write operations, the write operation has no effect; therefore, TOF remains set indicating an overflow has occurred. In this case, a TOF interrupt request is not lost due to the clearing sequence for a previous TOF. 0 FTM counter has not overflowed. 1 FTM counter has overflowed.

Table continues on the next page...

FTMx_SC field descriptions (continued)

Field	Description
6 TOIE	Timer Overflow Interrupt Enable Enables FTM overflow interrupts. 0 Disable TOF interrupts. Use software polling. 1 Enable TOF interrupts. An interrupt is generated when TOF equals one.
5 CPWMS	Center-Aligned PWM Select Selects CPWM mode. This mode configures the FTM to operate in Up-Down Counting mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 FTM counter operates in Up Counting mode. 1 FTM counter operates in Up-Down Counting mode.
4–3 CLKS	Clock Source Selection Selects one of the three FTM counter clock sources. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00 No clock selected. This in effect disables the FTM counter. 01 System clock 10 Fixed frequency clock 11 External clock
2–0 PS	Prescale Factor Selection Selects one of 8 division factors for the clock source selected by CLKS. The new prescaler factor affects the clock source on the next system clock cycle after the new value is updated into the register bits. This field is write protected. It can be written only when MODE[WPDIS] = 1. 000 Divide by 1 001 Divide by 2 010 Divide by 4 011 Divide by 8 100 Divide by 16 101 Divide by 32 110 Divide by 64 111 Divide by 128

39.3.4 Counter (FTMx_CNT)

The CNT register contains the FTM counter value.

Reset clears the CNT register. Writing any value to COUNT updates the counter with its initial value, CNTIN.

When BDM is active, the FTM counter is frozen. This is the value that you may read.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_CNT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 COUNT	Counter Value

39.3.5 Modulo (FTMx_MOD)

The Modulo register contains the modulo value for the FTM counter. After the FTM counter reaches the modulo value, the overflow flag (TOF) becomes set at the next clock, and the next value of FTM counter depends on the selected counting method; see Counter.

Writing to the MOD register latches the value into a buffer. The MOD register is updated with the value of its write buffer according to Registers updated from write buffers.

If FTMEN = 0, this write coherency mechanism may be manually reset by writing to the SC register whether BDM is active or not.

Initialize the FTM counter, by writing to CNT, before writing to the MOD register to avoid confusion about when the first counter overflow will occur.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																MOD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–0 MOD	Modulo Value

39.3.6 Channel (n) Status And Control (FTMx_C*n*SC)

CnSC contains the channel-interrupt-status flag and control bits used to configure the interrupt enable, channel configuration, and pin function.

Table 39-67. Mode, edge, and level selection

DECAPEN	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	X	X	XX	0	Pin not used for FTM—revert the channel pin to general purpose I/O or other peripheral control	
0	0	0	0	1	Input Capture	Capture on Rising Edge Only
				10		Capture on Falling Edge Only
				11		Capture on Rising or Falling Edge
			1	1	Output Compare	Toggle Output on match
				10		Clear Output on match
				11		Set Output on match
			1X	10	Edge-Aligned PWM	High-true pulses (clear Output on match)
				X1		Low-true pulses (set Output on match)
		1	XX	10	Center-Aligned PWM	High-true pulses (clear Output on match-up)
				X1		Low-true pulses (set Output on match-up)
	1	0	XX	10	Combine PWM	High-true pulses (set on channel (n) match, and clear on channel (n+1) match)
				X1		Low-true pulses (clear on channel (n) match, and set on channel (n +1) match)

Table continues on the next page...

Table 39-67. Mode, edge, and level selection (continued)

DECAPEN	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
1	0	0	X0	See the following table (Table 39-8).	Dual Edge Capture	One-Shot Capture mode
			X1			Continuous Capture mode

Table 39-68. Dual Edge Capture mode — edge polarity selection

ELSnB	ELSnA	Channel Port Enable	Detected Edges
0	0	Disabled	No edge
0	1	Enabled	Rising edge
1	0	Enabled	Falling edge
1	1	Enabled	Rising and falling edges

Address: Base address + Ch offset + (8d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CHF	CHIE	MSB	MSA	ELSB	ELSA	0	DMA
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_*Cn*SC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CHF	Channel Flag Set by hardware when an event occurs on the channel. CHF is cleared by reading the CSC register while CHnF is set and then writing a 0 to the CHF bit. Writing a 1 to CHF has no effect. If another event occurs between the read and write operations, the write operation has no effect; therefore, CHF remains set indicating an event has occurred. In this case a CHF interrupt request is not lost due to the clearing sequence for a previous CHF. 0 No channel event has occurred. 1 A channel event has occurred.
6 CHIE	Channel Interrupt Enable Enables channel interrupts.

Table continues on the next page...

FTMx_CnSC field descriptions (continued)

Field	Description
	0 Disable channel interrupts. Use software polling. 1 Enable channel interrupts.
5 MSB	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. See Table 39-7. This field is write protected. It can be written only when MODE[WPDIS] = 1.
4 MSA	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. See Table 39-7. This field is write protected. It can be written only when MODE[WPDIS] = 1.
3 ELSB	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. See Table 39-7. This field is write protected. It can be written only when MODE[WPDIS] = 1.
2 ELSA	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. See Table 39-7. This field is write protected. It can be written only when MODE[WPDIS] = 1.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DMA	DMA Enable Enables DMA transfers for the channel. 0 Disable DMA transfers. 1 Enable DMA transfers.

39.3.7 Channel (n) Value (FTMx_C*n*V)

These registers contain the captured FTM counter value for the input modes or the match value for the output modes.

In Input Capture, Capture Test, and Dual Edge Capture modes, any write to a CnV register is ignored.

In output modes, writing to a CnV register latches the value into a buffer. A CnV register is updated with the value of its write buffer according to Registers updated from write buffers.

If FTMEN = 0, this write coherency mechanism may be manually reset by writing to the CnSC register whether BDM mode is active or not.

Address: Base address + 10h offset + (8d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																VAL															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_C*n*V field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 VAL	Channel Value Captured FTM counter value of the input modes or the match value for the output modes

39.3.8 Counter Initial Value (FTMx_CNTIN)

The Counter Initial Value register contains the initial value for the FTM counter.

Writing to the CNTIN register latches the value into a buffer. The CNTIN register is updated with the value of its write buffer according to Registers updated from write buffers.

When the FTM clock is initially selected, by writing a non-zero value to the CLKS bits, the FTM counter starts with the value 0x0000. To avoid this behavior, before the first write to select the FTM clock, write the new value to the the CNTIN register and then initialize the FTM counter by writing any value to the CNT register.

Address: Base address + 4Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																INIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_CNTIN field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–0 INIT	Initial Value Of The FTM Counter

39.3.9 Capture And Compare Status (FTMx_STATUS)

The STATUS register contains a copy of the status flag CHnF bit in CnSC for each FTM channel for software convenience.

Each CHnF bit in STATUS is a mirror of CHnF bit in CnSC. All CHnF bits can be checked using only one read of STATUS. All CHnF bits can be cleared by reading STATUS followed by writing 0x00 to STATUS.

Hardware sets the individual channel flags when an event occurs on the channel. CHnF is cleared by reading STATUS while CHnF is set and then writing a 0 to the CHnF bit. Writing a 1 to CHnF has no effect.

If another event occurs between the read and write operations, the write operation has no effect; therefore, CHnF remains set indicating an event has occurred. In this case, a CHnF interrupt request is not lost due to the clearing sequence for a previous CHnF.

NOTE

The STATUS register should be used only in Combine mode.

Address: Base address + 50h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7F	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W									0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_STATUS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7F	Channel 7 Flag See the register description.

Table continues on the next page...

FTMx_STATUS field descriptions (continued)

Field	Description
	0 No channel event has occurred. 1 A channel event has occurred.
6 CH6F	Channel 6 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
5 CH5F	Channel 5 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
4 CH4F	Channel 4 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
3 CH3F	Channel 3 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
2 CH2F	Channel 2 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
1 CH1F	Channel 1 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
0 CH0F	Channel 0 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.

39.3.10 Features Mode Selection (FTMx_MODE)

This register contains the global enable bit for FTM-specific features and the control bits used to configure:

- Fault control mode and interrupt
- Capture Test mode
- PWM synchronization
- Write protection
- Channel output initialization

These controls relate to all channels within this module.

Address: Base address + 54h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								FAULTIE	FAULTM		CAPTEST	PWMSYNC	WPDIS	INIT	FTMEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

FTMx_MODE field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 FAULTIE	Fault Interrupt Enable Enables the generation of an interrupt when a fault is detected by FTM and the FTM fault control is enabled. 0 Fault control interrupt is disabled. 1 Fault control interrupt is enabled.
6–5 FAULTM	Fault Control Mode Defines the FTM fault control mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00 Fault control is disabled for all channels. 01 Fault control is enabled for even channels only (channels 0, 2, 4, and 6), and the selected mode is the manual fault clearing. 10 Fault control is enabled for all channels, and the selected mode is the manual fault clearing. 11 Fault control is enabled for all channels, and the selected mode is the automatic fault clearing.
4 CAPTEST	Capture Test Mode Enable Enables the capture test mode. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_MODE field descriptions (continued)

Field	Description
	0 Capture test mode is disabled. 1 Capture test mode is enabled.
3 PWMSYNC	PWM Synchronization Mode Selects which triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. See PWM synchronization. The PWMSYNC bit configures the synchronization when SYNCMODE is zero. 0 No restrictions. Software and hardware triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. 1 Software trigger can only be used by MOD and CnV synchronization, and hardware triggers can only be used by OUTMASK and FTM counter synchronization.
2 WPDIS	Write Protection Disable When write protection is enabled (WPDIS = 0), write protected bits cannot be written. When write protection is disabled (WPDIS = 1), write protected bits can be written. The WPDIS bit is the negation of the WPEN bit. WPDIS is cleared when 1 is written to WPEN. WPDIS is set when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPDIS has no effect. 0 Write protection is enabled. 1 Write protection is disabled.
1 INIT	Initialize The Channels Output When a 1 is written to INIT bit the channels output is initialized according to the state of their corresponding bit in the OUTINIT register. Writing a 0 to INIT bit has no effect. The INIT bit is always read as 0.
0 FTMEN	FTM Enable This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Only the TPM-compatible registers (first set of registers) can be used without any restriction. Do not use the FTM-specific registers. 1 All registers including the FTM-specific registers (second set of registers) are available for use with no restrictions.

39.3.11 Synchronization (FTMx_SYNC)

This register configures the PWM synchronization.

A synchronization event can perform the synchronized update of MOD, CV, and OUTMASK registers with the value of their write buffer and the FTM counter initialization.

NOTE

The software trigger, SWSYNC bit, and hardware triggers TRIG0, TRIG1, and TRIG2 bits have a potential conflict if used together when SYNCMODE = 0. Use only hardware or software triggers but not both at the same time, otherwise unpredictable behavior is likely to happen.

The selection of the loading point, CNTMAX and CNTMIN bits, is intended to provide the update of MOD, CNTIN, and CnV registers across all enabled channels simultaneously. The use of the loading point selection together with SYNCMODE = 0 and hardware trigger selection, TRIG0, TRIG1, or TRIG2 bits, is likely to result in unpredictable behavior.

The synchronization event selection also depends on the PWMSYNC (MODE register) and SYNCMODE (SYNCONF register) bits. See PWM synchronization.

Address: Base address + 58h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SWSYNC	TRIG2	TRIG1	TRIG0	SYNCHOM	REINIT	CNTMAX	CNTMIN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SYNC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 SWSYNC	PWM Synchronization Software Trigger Selects the software trigger as the PWM synchronization trigger. The software trigger happens when a 1 is written to SWSYNC bit. 0 Software trigger is not selected. 1 Software trigger is selected.
6 TRIG2	PWM Synchronization Hardware Trigger 2

Table continues on the next page...

FTMx_SYNC field descriptions (continued)

Field	Description
	Enables hardware trigger 2 to the PWM synchronization. Hardware trigger 2 happens when a rising edge is detected at the trigger 2 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
5 TRIG1	PWM Synchronization Hardware Trigger 1 Enables hardware trigger 1 to the PWM synchronization. Hardware trigger 1 happens when a rising edge is detected at the trigger 1 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
4 TRIG0	PWM Synchronization Hardware Trigger 0 Enables hardware trigger 0 to the PWM synchronization. Hardware trigger 0 happens when a rising edge is detected at the trigger 0 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
3 SYNCHOM	Output Mask Synchronization Selects when the OUTMASK register is updated with the value of its buffer. 0 OUTMASK register is updated with the value of its buffer in all rising edges of the system clock. 1 OUTMASK register is updated with the value of its buffer only by the PWM synchronization.
2 REINIT	FTM Counter Reinitialization By Synchronization (FTM counter synchronization) Determines if the FTM counter is reinitialized when the selected trigger for the synchronization is detected. The REINIT bit configures the synchronization when SYNCMODE is zero. 0 FTM counter continues to count normally. 1 FTM counter is updated with its initial value when the selected trigger is detected.
1 CNTMAX	Maximum Loading Point Enable Selects the maximum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMAX is one, the selected loading point is when the FTM counter reaches its maximum value (MOD register). 0 The maximum loading point is disabled. 1 The maximum loading point is enabled.
0 CNTMIN	Minimum Loading Point Enable Selects the minimum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMIN is one, the selected loading point is when the FTM counter reaches its minimum value (CNTIN register). 0 The minimum loading point is disabled. 1 The minimum loading point is enabled.

39.3.12 Initial State For Channels Output (FTMx_OUTINIT)

Address: Base address + 5Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7OI	CH6OI	CH5OI	CH4OI	CH3OI	CH2OI	CH1OI	CH0OI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_OUTINIT field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7OI	Channel 7 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
6 CH6OI	Channel 6 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
5 CH5OI	Channel 5 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
4 CH4OI	Channel 4 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
3 CH3OI	Channel 3 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs.

Table continues on the next page...

FTMx_OUTINIT field descriptions (continued)

Field	Description
	0 The initialization value is 0. 1 The initialization value is 1.
2 CH2OI	Channel 2 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
1 CH1OI	Channel 1 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
0 CH0OI	Channel 0 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.

39.3.13 Output Mask (FTMx_OUTMASK)

This register provides a mask for each FTM channel. The mask of a channel determines if its output responds, that is, it is masked or not, when a match occurs. This feature is used for BLDC control where the PWM signal is presented to an electric motor at specific times to provide electronic commutation.

Any write to the OUTMASK register, stores the value in its write buffer. The register is updated with the value of its write buffer according to PWM synchronization.

Address: Base address + 60h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7OM	CH6OM	CH5OM	CH4OM	CH3OM	CH2OM	CH1OM	CH0OM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_OUTMASK field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7OM	Channel 7 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
6 CH6OM	Channel 6 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
5 CH5OM	Channel 5 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
4 CH4OM	Channel 4 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
3 CH3OM	Channel 3 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
2 CH2OM	Channel 2 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
1 CH1OM	Channel 1 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
0 CH0OM	Channel 0 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.

39.3.14 Function For Linked Channels (FTMx_COMBINE)

This register contains the control bits used to configure the fault control, synchronization, deadtime insertion, Dual Edge Capture mode, Complementary, and Combine mode for each pair of channels (n) and (n+1), where n equals 0, 2, 4, and 6.

Address: Base address + 64h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	FAULTEN3	SYNCEN3	DTEN3	DECAP3	DECAPEN3	COMP3	COMBINE3	0	FAULTEN2	SYNCEN2	DTEN2	DECAP2	DECAPEN2	COMP2	COMBINE2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	FAULTEN1	SYNCEN1	DTEN1	DECAP1	DECAPEN1	COMP1	COMBINE1	0	FAULTEN0	SYNCEN0	DTEN0	DECAP0	DECAPEN0	COMP0	COMBINE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_COMBINE field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 FAULTEN3	Fault Control Enable For n = 6 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
29 SYNCEN3	Synchronization Enable For n = 6 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
28 DTEN3	Deadtime Enable For n = 6 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
27 DECAP3	Dual Edge Capture Mode Captures For n = 6 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when FTMEN = 1 and DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
26 DECAPEN3	Dual Edge Capture Mode Enable For n = 6 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 39-7. This field applies only when FTMEN = 1. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
25 COMP3	Complement Of Channel (n) for n = 6 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
24 COMBINE3	Combine Channels For n = 6 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 FAULTEN2	Fault Control Enable For n = 4 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
21 SYNCEN2	Synchronization Enable For n = 4 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
20 DTEN2	Deadtime Enable For n = 4 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
19 DECAP2	Dual Edge Capture Mode Captures For n = 4 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when FTMEN = 1 and DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
18 DECAPEN2	Dual Edge Capture Mode Enable For n = 4 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 39-7. This field applies only when FTMEN = 1. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
17 COMP2	Complement Of Channel (n) For n = 4 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
16 COMBINE2	Combine Channels For n = 4 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 FAULTEN1	Fault Control Enable For n = 2 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
13 SYNCEN1	Synchronization Enable For n = 2 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
12 DTEN1	Deadtime Enable For n = 2 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
11 DECAP1	Dual Edge Capture Mode Captures For n = 2 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when FTMEN = 1 and DECAPEN = 1. DECAP bit is cleared automatically by hardware if Dual Edge Capture – One-Shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
10 DECAPEN1	Dual Edge Capture Mode Enable For n = 2 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 39-7. This field applies only when FTMEN = 1. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
9 COMP1	Complement Of Channel (n) For n = 2 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
8 COMBINE1	Combine Channels For n = 2 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FAULTEN0	Fault Control Enable For n = 0 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
5 SYNCEN0	Synchronization Enable For n = 0 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
4 DTEN0	Deadtime Enable For n = 0 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
3 DECAP0	Dual Edge Capture Mode Captures For n = 0 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when FTMEN = 1 and DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
2 DECAPEN0	Dual Edge Capture Mode Enable For n = 0 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 39-7. This field applies only when FTMEN = 1. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
1 COMP0	Complement Of Channel (n) For n = 0 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
0 COMBINE0	Combine Channels For n = 0 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.

39.3.15 Deadtime Insertion Control (FTMx_DEADTIME)

This register selects the deadtime prescaler factor and deadtime value. All FTM channels use this clock prescaler and this deadtime value for the deadtime insertion.

Address: Base address + 68h offset

Bit	31–8	7	6	5	4	3	2	1	0
R	0	DTPS		DTVAL					
W									
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0	0	0	0	0	0	0

FTMx_DEADTIME field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 DTPS	Deadtime Prescaler Value Selects the division factor of the system clock. This prescaled clock is used by the deadtime counter. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0x Divide the system clock by 1. 10 Divide the system clock by 4. 11 Divide the system clock by 16.
5–0 DTVAL	Deadtime Value Selects the deadtime insertion value for the deadtime counter. The deadtime counter is clocked by a scaled version of the system clock. See the description of DTPS. Deadtime insert value = (DTPS × DTVAL). DTVAL selects the number of deadtime counts inserted as follows: When DTVAL is 0, no counts are inserted. When DTVAL is 1, 1 count is inserted. When DTVAL is 2, 2 counts are inserted. This pattern continues up to a possible 63 counts. This field is write protected. It can be written only when MODE[WPDIS] = 1.

39.3.16 FTM External Trigger (FTMx_EXTTRIG)

This register:

- Indicates when a channel trigger was generated
- Enables the generation of a trigger when the FTM counter is equal to its initial value
- Selects which channels are used in the generation of the channel triggers

Several channels can be selected to generate multiple triggers in one PWM period.

Channels 6 and 7 are not used to generate channel triggers.

Address: Base address + 6Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TRIGF	INITTRIGEN	CH1TRIG	CH0TRIG	CH5TRIG	CH4TRIG	CH3TRIG	CH2TRIG
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_EXTTRIG field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 TRIGF	Channel Trigger Flag Set by hardware when a channel trigger is generated. Clear TRIGF by reading EXTTRIG while TRIGF is set and then writing a 0 to TRIGF. Writing a 1 to TRIGF has no effect. If another channel trigger is generated before the clearing sequence is completed, the sequence is reset so TRIGF remains set after the clear sequence is completed for the earlier TRIGF. 0 No channel trigger was generated. 1 A channel trigger was generated.
6 INITTRIGEN	Initialization Trigger Enable Enables the generation of the trigger when the FTM counter is equal to the CNTIN register. 0 The generation of initialization trigger is disabled. 1 The generation of initialization trigger is enabled.
5 CH1TRIG	Channel 1 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
4 CH0TRIG	Channel 0 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
3 CH5TRIG	Channel 5 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
2 CH4TRIG	Channel 4 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
1 CH3TRIG	Channel 3 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
0 CH2TRIG	Channel 2 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register.

Table continues on the next page...

FTMx_EXTTRIG field descriptions (continued)

Field	Description
	0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.

39.3.17 Channels Polarity (FTMx_POL)

This register defines the output polarity of the FTM channels.

NOTE

The safe value that is driven in a channel output when the fault control is enabled and a fault condition is detected is the inactive state of the channel. That is, the safe value of a channel is the value of its POL bit.

Address: Base address + 70h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_POL field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 POL7	Channel 7 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
6 POL6	Channel 6 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
5 POL5	Channel 5 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_POL field descriptions (continued)

Field	Description
	0 The channel polarity is active high. 1 The channel polarity is active low.
4 POL4	Channel 4 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
3 POL3	Channel 3 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
2 POL2	Channel 2 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
1 POL1	Channel 1 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
0 POL0	Channel 0 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.

39.3.18 Fault Mode Status (FTMx_FMS)

This register contains the fault detection flags, write protection enable bit, and the logic OR of the enabled fault inputs.

Address: Base address + 74h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								FAULTF	WPEN	FAULTIN	0	FAULTF3	FAULTF2	FAULTF1	FAULTF0
W									0				0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FMS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 FAULTF	Fault Detection Flag Represents the logic OR of the individual FAULTFj bits where j = 3, 2, 1, 0. Clear FAULTF by reading the FMS register while FAULTF is set and then writing a 0 to FAULTF while there is no existing fault condition at the enabled fault inputs. Writing a 1 to FAULTF has no effect. If another fault condition is detected in an enabled fault input before the clearing sequence is completed, the sequence is reset so FAULTF remains set after the clearing sequence is completed for the earlier fault condition. FAULTF is also cleared when FAULTFj bits are cleared individually. 0 No fault condition was detected. 1 A fault condition was detected.

Table continues on the next page...

FTMx_FMS field descriptions (continued)

Field	Description
6 WPEN	Write Protection Enable The WPEN bit is the negation of the WPDIS bit. WPEN is set when 1 is written to it. WPEN is cleared when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPEN has no effect. 0 Write protection is disabled. Write protected bits can be written. 1 Write protection is enabled. Write protected bits cannot be written.
5 FAULTIN	Fault Inputs Represents the logic OR of the enabled fault inputs after their filter (if their filter is enabled) when fault control is enabled. 0 The logic OR of the enabled fault inputs is 0. 1 The logic OR of the enabled fault inputs is 1.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 FAULTF3	Fault Detection Flag 3 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF3 by reading the FMS register while FAULTF3 is set and then writing a 0 to FAULTF3 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF3 has no effect. FAULTF3 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF3 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
2 FAULTF2	Fault Detection Flag 2 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF2 by reading the FMS register while FAULTF2 is set and then writing a 0 to FAULTF2 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF2 has no effect. FAULTF2 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF2 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
1 FAULTF1	Fault Detection Flag 1 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF1 by reading the FMS register while FAULTF1 is set and then writing a 0 to FAULTF1 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF1 has no effect. FAULTF1 bit is also cleared when FAULTF bit is cleared.

Table continues on the next page...

FTMx_FMS field descriptions (continued)

Field	Description
	If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF1 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
0 FAULTF0	Fault Detection Flag 0 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF0 by reading the FMS register while FAULTF0 is set and then writing a 0 to FAULTF0 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF0 has no effect. FAULTF0 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF0 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.

39.3.19 Input Capture Filter Control (FTMx_FILTER)

This register selects the filter value for the inputs of channels.

Channels 4, 5, 6 and 7 do not have an input filter.

NOTE

Writing to the FILTER register has immediate effect and must be done only when the channels 0, 1, 2, and 3 are not in input modes. Failure to do this could result in a missing valid signal.

Address: Base address + 78h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																CH3FVAL				CH2FVAL				CH1FVAL				CH0FVAL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FILTER field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–12 CH3FVAL	Channel 3 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.

Table continues on the next page...

FTMx_FILTER field descriptions (continued)

Field	Description
11–8 CH2FVAL	Channel 2 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
7–4 CH1FVAL	Channel 1 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
3–0 CH0FVAL	Channel 0 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.

39.3.20 Fault Control (FTMx_FLTCTRL)

This register selects the filter value for the fault inputs, enables the fault inputs and the fault inputs filter.

Address: Base address + 7Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FFVAL				FFLTR3EN	FFLTR2EN	FFLTR1EN	FFLTR0EN	FAULT3EN	FAULT2EN	FAULT1EN	FAULT0EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FLTCTRL field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 FFVAL	Fault Input Filter Selects the filter value for the fault inputs. The fault filter is disabled when the value is zero.

Table continues on the next page...

FTMx_FLTCTRL field descriptions (continued)

Field	Description
	NOTE: Writing to this field has immediate effect and must be done only when the fault control or all fault inputs are disabled. Failure to do this could result in a missing fault detection.
7 FFLTR3EN	Fault Input 3 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
6 FFLTR2EN	Fault Input 2 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
5 FFLTR1EN	Fault Input 1 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
4 FFLTR0EN	Fault Input 0 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
3 FAULT3EN	Fault Input 3 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.
2 FAULT2EN	Fault Input 2 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.
1 FAULT1EN	Fault Input 1 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_FLTCTRL field descriptions (continued)

Field	Description
	0 Fault input is disabled. 1 Fault input is enabled.
0 FAULT0EN	Fault Input 0 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.

39.3.21 Quadrature Decoder Control And Status (FTMx_QDCTRL)

This register has the control and status bits for the Quadrature Decoder mode.

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PHAFLTREN	PHBFLTREN	PHAPOL	PHBPOL	QUADMODE	QUADIR	TOFDIR	QUADEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_QDCTRL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 PHAFLTREN	Phase A Input Filter Enable Enables the filter for the quadrature decoder phase A input. The filter value for the phase A input is defined by the CH0FVAL field of FILTER. The phase A filter is also disabled when CH0FVAL is zero. 0 Phase A input filter is disabled. 1 Phase A input filter is enabled.
6 PHBFLTREN	Phase B Input Filter Enable Enables the filter for the quadrature decoder phase B input. The filter value for the phase B input is defined by the CH1FVAL field of FILTER. The phase B filter is also disabled when CH1FVAL is zero. 0 Phase B input filter is disabled. 1 Phase B input filter is enabled.
5 PHAPOL	Phase A Input Polarity Selects the polarity for the quadrature decoder phase A input. 0 Normal polarity. Phase A input signal is not inverted before identifying the rising and falling edges of this signal. 1 Inverted polarity. Phase A input signal is inverted before identifying the rising and falling edges of this signal.
4 PHBPOL	Phase B Input Polarity Selects the polarity for the quadrature decoder phase B input. 0 Normal polarity. Phase B input signal is not inverted before identifying the rising and falling edges of this signal. 1 Inverted polarity. Phase B input signal is inverted before identifying the rising and falling edges of this signal.
3 QUADMODE	Quadrature Decoder Mode Selects the encoding mode used in the Quadrature Decoder mode. 0 Phase A and phase B encoding mode. 1 Count and direction encoding mode.
2 QUADIR	FTM Counter Direction In Quadrature Decoder Mode Indicates the counting direction. 0 Counting direction is decreasing (FTM counter decrement). 1 Counting direction is increasing (FTM counter increment).
1 TOFDIR	Timer Overflow Direction In Quadrature Decoder Mode Indicates if the TOF bit was set on the top or the bottom of counting. 0 TOF bit was set on the bottom of counting. There was an FTM counter decrement and FTM counter changes from its minimum value (CNTIN register) to its maximum value (MOD register). 1 TOF bit was set on the top of counting. There was an FTM counter increment and FTM counter changes from its maximum value (MOD register) to its minimum value (CNTIN register).

Table continues on the next page...

FTMx_QDCTRL field descriptions (continued)

Field	Description
0 QUADEN	Quadrature Decoder Mode Enable Enables the Quadrature Decoder mode. In this mode, the phase A and B input signals control the FTM counter direction. The Quadrature Decoder mode has precedence over the other modes. See Table 39-7. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Quadrature Decoder mode is disabled. 1 Quadrature Decoder mode is enabled.

39.3.22 Configuration (FTMx_CONF)

This register selects the number of times that the FTM counter overflow should occur before the TOF bit to be set, the FTM behavior in BDM modes, the use of an external global time base, and the global time base signal generation.

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					GTBEOUT	GTBEEN	0	BDMMODE		0	NUMTOF				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_CONF field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 GTBEOUT	Global Time Base Output Enables the global time base signal generation to other FTMs. 0 A global time base signal generation is disabled. 1 A global time base signal generation is enabled.
9 GTBEEN	Global Time Base Enable

Table continues on the next page...

FTMx_CONF field descriptions (continued)

Field	Description
	Configures the FTM to use an external global time base signal that is generated by another FTM. 0 Use of an external global time base is disabled. 1 Use of an external global time base is enabled.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 BDMMODE	BDM Mode Selects the FTM behavior in BDM mode. See BDM mode.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 NUMTOF	TOF Frequency Selects the ratio between the number of counter overflows to the number of times the TOF bit is set. NUMTOF = 0: The TOF bit is set for each counter overflow. NUMTOF = 1: The TOF bit is set for the first counter overflow but not for the next overflow. NUMTOF = 2: The TOF bit is set for the first counter overflow but not for the next 2 overflows. NUMTOF = 3: The TOF bit is set for the first counter overflow but not for the next 3 overflows. This pattern continues up to a maximum of 31.

39.3.23 FTM Fault Input Polarity (FTMx_FLTPOL)

This register defines the fault inputs polarity.

Address: Base address + 88h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												FLT3POL	FLT2POL	FLT1POL	FLT0POL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FLTPOL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

FTMx_FLTPOL field descriptions (continued)

Field	Description
3 FLT3POL	Fault Input 3 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A one at the fault input indicates a fault. 1 The fault input polarity is active low. A zero at the fault input indicates a fault.
2 FLT2POL	Fault Input 2 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A one at the fault input indicates a fault. 1 The fault input polarity is active low. A zero at the fault input indicates a fault.
1 FLT1POL	Fault Input 1 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A one at the fault input indicates a fault. 1 The fault input polarity is active low. A zero at the fault input indicates a fault.
0 FLT0POL	Fault Input 0 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A one at the fault input indicates a fault. 1 The fault input polarity is active low. A zero at the fault input indicates a fault.

39.3.24 Synchronization Configuration (FTMx_SYNCONF)

This register selects the PWM synchronization configuration, SWOCTRL, INVCTRL and CNTIN registers synchronization, if FTM clears the TRIGj bit, where j = 0, 1, 2, when the hardware trigger j is detected.

Address: Base address + 8Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											HWSOC	HWINVC	HWOM	HWWRBUF	HWRSTCNT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SWSOC	SWINVC	SWOM	SWWRBUF	SWRSTCNT	SYNCMODE	0	SWOC	INVC	0	CNTINC	0	HWTRIGMODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SYNCONF field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 HWSOC	Software output control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the SWOCTRL register synchronization. 1 A hardware trigger activates the SWOCTRL register synchronization.
19 HWINVC	Inverting control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the INVCTRL register synchronization. 1 A hardware trigger activates the INVCTRL register synchronization.
18 HWOM	Output mask synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the OUTMASK register synchronization. 1 A hardware trigger activates the OUTMASK register synchronization.
17 HWWRBUF	MOD, CNTIN, and CV registers synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate MOD, CNTIN, and CV registers synchronization. 1 A hardware trigger activates MOD, CNTIN, and CV registers synchronization.
16 HWRSTCNT	FTM counter synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the FTM counter synchronization. 1 A hardware trigger activates the FTM counter synchronization.

Table continues on the next page...

FTMx_SYNCONF field descriptions (continued)

Field	Description
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 SWSOC	Software output control synchronization is activated by the software trigger. 0 The software trigger does not activate the SWOCTRL register synchronization. 1 The software trigger activates the SWOCTRL register synchronization.
11 SWINVC	Inverting control synchronization is activated by the software trigger. 0 The software trigger does not activate the INVCTRL register synchronization. 1 The software trigger activates the INVCTRL register synchronization.
10 SWOM	Output mask synchronization is activated by the software trigger. 0 The software trigger does not activate the OUTMASK register synchronization. 1 The software trigger activates the OUTMASK register synchronization.
9 SWWRBUF	MOD, CNTIN, and CV registers synchronization is activated by the software trigger. 0 The software trigger does not activate MOD, CNTIN, and CV registers synchronization. 1 The software trigger activates MOD, CNTIN, and CV registers synchronization.
8 SWRSTCNT	FTM counter synchronization is activated by the software trigger. 0 The software trigger does not activate the FTM counter synchronization. 1 The software trigger activates the FTM counter synchronization.
7 SYNCMODE	Synchronization Mode Selects the PWM Synchronization mode. 0 Legacy PWM synchronization is selected. 1 Enhanced PWM synchronization is selected.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 SWOC	SWOCTRL Register Synchronization 0 SWOCTRL register is updated with its buffer value at all rising edges of system clock. 1 SWOCTRL register is updated with its buffer value by the PWM synchronization.
4 INVC	INVCTRL Register Synchronization 0 INVCTRL register is updated with its buffer value at all rising edges of system clock. 1 INVCTRL register is updated with its buffer value by the PWM synchronization.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CNTINC	CNTIN Register Synchronization 0 CNTIN register is updated with its buffer value at all rising edges of system clock. 1 CNTIN register is updated with its buffer value by the PWM synchronization.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 HWTRIGMODE	Hardware Trigger Mode

Table continues on the next page...

FTMx_SYNCONF field descriptions (continued)

Field	Description
	0 FTM clears the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2. 1 FTM does not clear the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2.

39.3.25 FTM Inverting Control (FTMx_INVCTRL)

This register controls when the channel (n) output becomes the channel (n+1) output, and channel (n+1) output becomes the channel (n) output. Each INVmEN bit enables the inverting operation for the corresponding pair channels m.

This register has a write buffer. The INVmEN bit is updated by the INVCTRL register synchronization.

Address: Base address + 90h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												INV3EN	INV2EN	INV1EN	INV0EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_INVCTRL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 INV3EN	Pair Channels 3 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
2 INV2EN	Pair Channels 2 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
1 INV1EN	Pair Channels 1 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.

Table continues on the next page...

FTMx_INVCTRL field descriptions (continued)

Field	Description
0 INV0EN	Pair Channels 0 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.

39.3.26 FTM Software Output Control (FTMx_SWOCTRL)

This register enables software control of channel (n) output and defines the value forced to the channel (n) output:

- The CHnOC bits enable the control of the corresponding channel (n) output by software.
- The CHnOCV bits select the value that is forced at the corresponding channel (n) output.

This register has a write buffer. The fields are updated by the SWOCTRL register synchronization.

Address: Base address + 94h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	CH7OCV	CH6OCV	CH5OCV	CH4OCV	CH3OCV	CH2OCV	CH1OCV	CH0OCV	CH7OC	CH6OC	CH5OC	CH4OC	CH3OC	CH2OC	CH1OC	CH0OC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SWOCTRL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 CH7OCV	Channel 7 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
14 CH6OCV	Channel 6 Software Output Control Value

Table continues on the next page...

FTMx_SWOCTRL field descriptions (continued)

Field	Description
	0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
13 CH5OCV	Channel 5 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
12 CH4OCV	Channel 4 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
11 CH3OCV	Channel 3 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
10 CH2OCV	Channel 2 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
9 CH1OCV	Channel 1 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
8 CH0OCV	Channel 0 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
7 CH7OC	Channel 7 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
6 CH6OC	Channel 6 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
5 CH5OC	Channel 5 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
4 CH4OC	Channel 4 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
3 CH3OC	Channel 3 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
2 CH2OC	Channel 2 Software Output Control Enable

Table continues on the next page...

FTMx_SWOCTRL field descriptions (continued)

Field	Description
	0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
1 CH1OC	Channel 1 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
0 CH0OC	Channel 0 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.

39.3.27 FTM PWM Load (FTMx_PWMLOAD)

Enables the loading of the MOD, CNTIN, C(n)V, and C(n+1)V registers with the values of their write buffers when the FTM counter changes from the MOD register value to its next value or when a channel (j) match occurs. A match occurs for the channel (j) when FTM counter = C(j)V.

Address: Base address + 98h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						LDOK	0	CH7SEL	CH6SEL	CH5SEL	CH4SEL	CH3SEL	CH2SEL	CH1SEL	CH0SEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_PWMLOAD field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 LDOK	Load Enable Enables the loading of the MOD, CNTIN, and CV registers with the values of their write buffers. 0 Loading updated values is disabled. 1 Loading updated values is enabled.

Table continues on the next page...

FTMx_PWMLOAD field descriptions (continued)

Field	Description
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7SEL	Channel 7 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
6 CH6SEL	Channel 6 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
5 CH5SEL	Channel 5 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
4 CH4SEL	Channel 4 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
3 CH3SEL	Channel 3 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
2 CH2SEL	Channel 2 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
1 CH1SEL	Channel 1 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
0 CH0SEL	Channel 0 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.

39.4 Functional description

The notation used in this document to represent the counters and the generation of the signals is shown in the following figure.

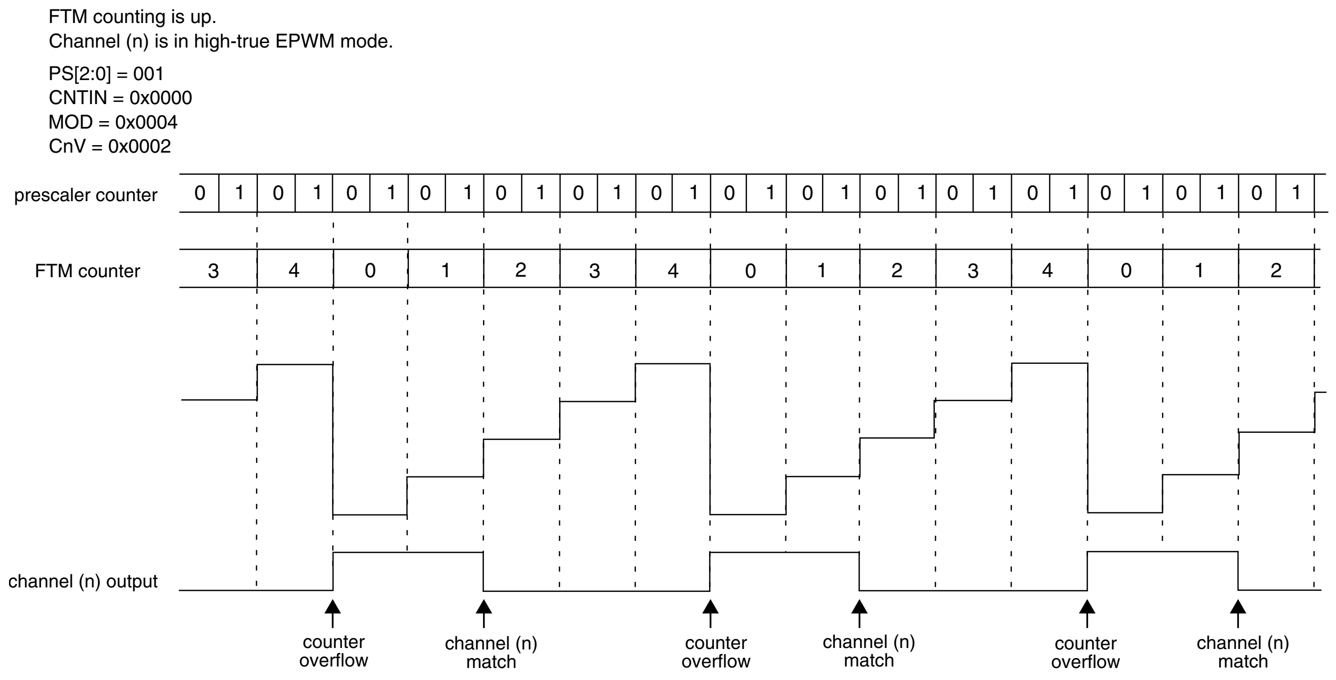


Figure 39-207. Notation used

39.4.1 Clock source

The FTM has only one clock domain: the system clock.

39.4.1.1 Counter clock source

The CLKS[1:0] bits in the SC register select one of three possible clock sources for the FTM counter or disable the FTM counter. After any MCU reset, CLKS[1:0] = 0:0 so no clock source is selected.

The CLKS[1:0] bits may be read or written at any time. Disabling the FTM counter by writing 0:0 to the CLKS[1:0] bits does not affect the FTM counter value or other registers.

The fixed frequency clock is an alternative clock source for the FTM counter that allows the selection of a clock other than the system clock or an external clock. This clock input is defined by chip integration. Refer to the chip specific documentation for further information. Due to FTM hardware implementation limitations, the frequency of the fixed frequency clock must not exceed 1/2 of the system clock frequency.

The external clock passes through a synchronizer clocked by the system clock to assure that counter transitions are properly aligned to system clock transitions.Therefore, to meet Nyquist criteria considering also jitter, the frequency of the external clock source must not exceed 1/4 of the system clock frequency.

39.4.2 Prescaler

The selected counter clock source passes through a prescaler that is a 7-bit counter. The value of the prescaler is selected by the PS[2:0] bits. The following figure shows an example of the prescaler counter and FTM counter.

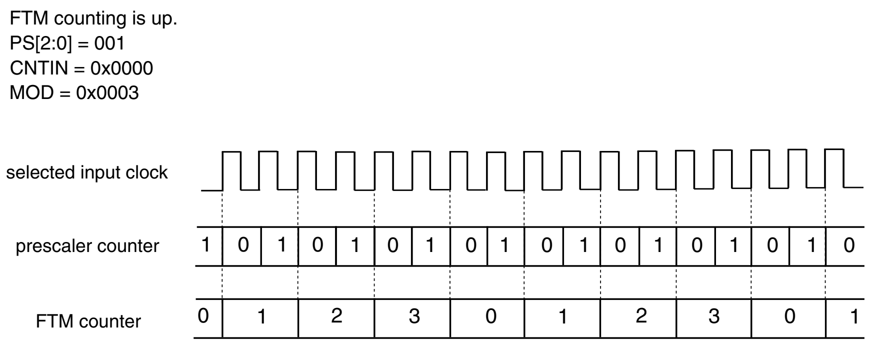


Figure 39-208. Example of the prescaler counter

39.4.3 Counter

The FTM has a 16-bit counter that is used by the channels either for input or output modes. The FTM counter clock is the selected clock divided by the prescaler.

The FTM counter has these modes of operation:

- Up counting
- Up-down counting
- Quadrature Decoder mode

39.4.3.1 Up counting

Up counting is selected when:

- QUADEN = 0, and
- CPWMS = 0

CNTIN defines the starting value of the count and MOD defines the final value of the count, see the following figure. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is reloaded with the value of CNTIN.

The FTM period when using up counting is (MOD – CNTIN + 0x0001) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to CNTIN.

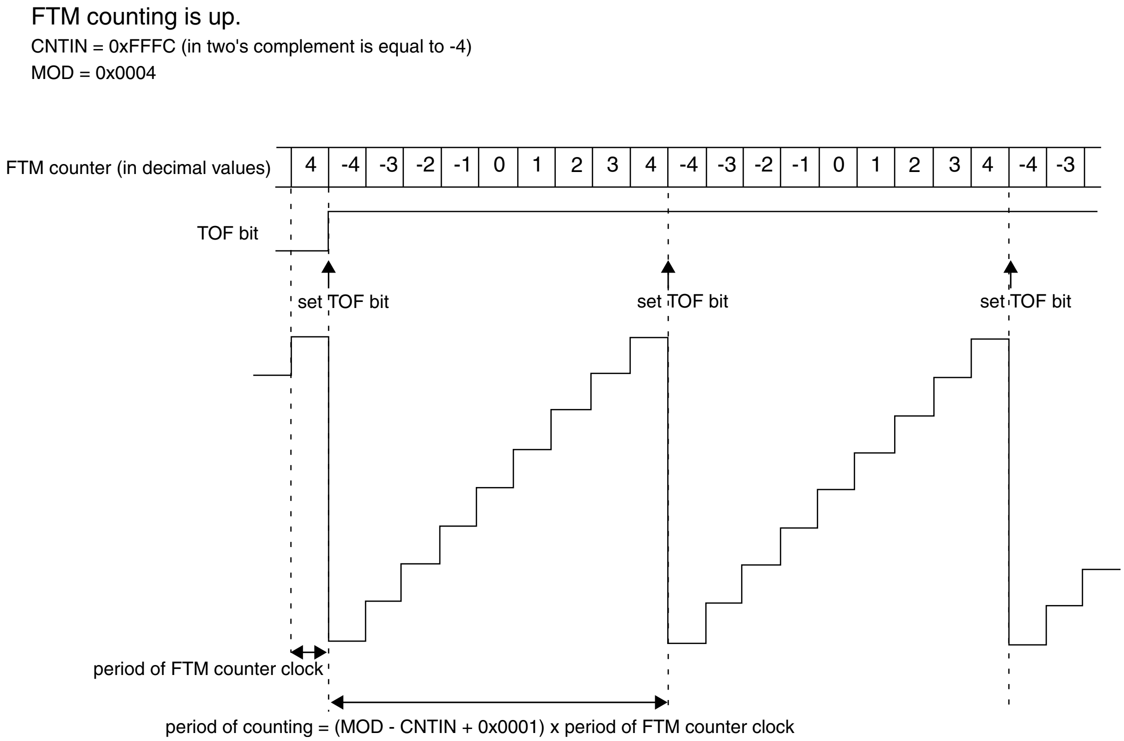


Figure 39-209. Example of FTM up and signed counting

Table 39-302. FTM counting based on CNTIN value

When	Then
CNTIN = 0x0000	The FTM counting is equivalent to TPM up counting, that is, up and unsigned counting. See the following figure.
CNTIN[15] = 1	The initial value of the FTM counter is a negative number in two's complement, so the FTM counting is up and signed.
CNTIN[15] = 0 and CNTIN ≠ 0x0000	The initial value of the FTM counter is a positive number, so the FTM counting is up and unsigned.

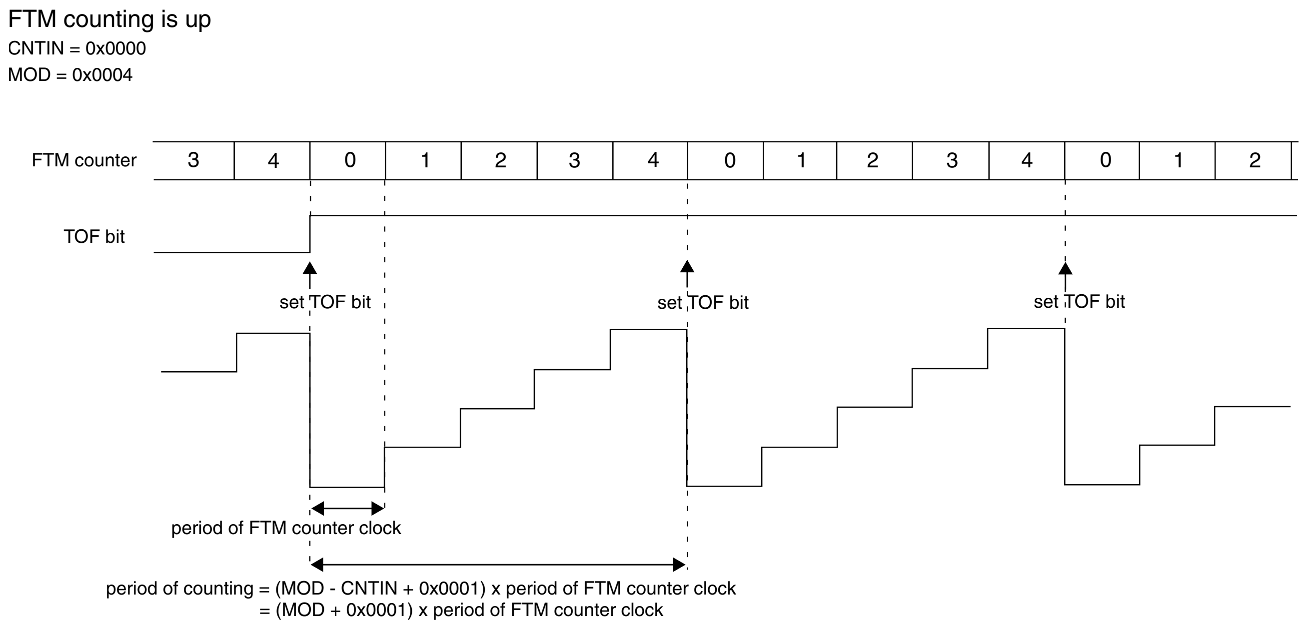


Figure 39-210. Example of FTM up counting with CNTIN = 0x0000

Note

- FTM operation is only valid when the value of the CNTIN register is less than the value of the MOD register, either in the unsigned counting or signed counting. It is the responsibility of the software to ensure that the values in the CNTIN and MOD registers meet this requirement. Any values of CNTIN and MOD that do not satisfy this criteria can result in unpredictable behavior.
- MOD = CNTIN is a redundant condition. In this case, the FTM counter is always equal to MOD and the TOF bit is set in each rising edge of the FTM counter clock.
- When MOD = 0x0000, CNTIN = 0x0000, for example after reset, and FTMEN = 1, the FTM counter remains stopped at 0x0000 until a non-zero value is written into the MOD or CNTIN registers.
- Setting CNTIN to be greater than the value of MOD is not recommended as this unusual setting may make the FTM operation difficult to comprehend. However, there is no restriction on this configuration, and an example is shown in the following figure.

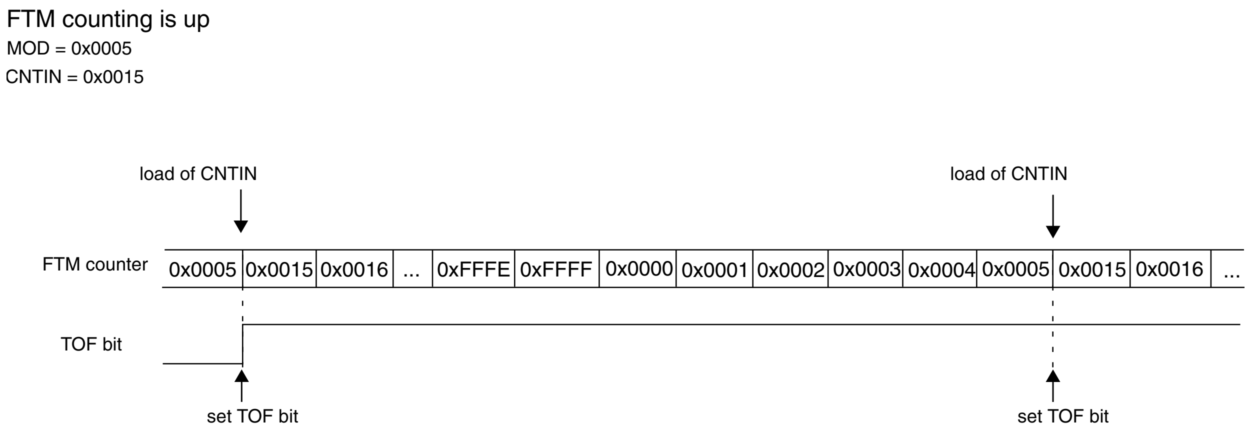


Figure 39-211. Example of up counting when the value of CNTIN is greater than the value of MOD

39.4.3.2 Up-down counting

Up-down counting is selected when:

- QUADEN = 0, and
- CPWMS = 1

CNTIN defines the starting value of the count and MOD defines the final value of the count. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is decremented until it returns to the value of CNTIN and the up-down counting restarts.

The FTM period when using up-down counting is 2 × (MOD – CNTIN) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to (MOD – 1).

If (CNTIN = 0x0000), the FTM counting is equivalent to TPM up-down counting, that is, up-down and unsigned counting. See the following figure.

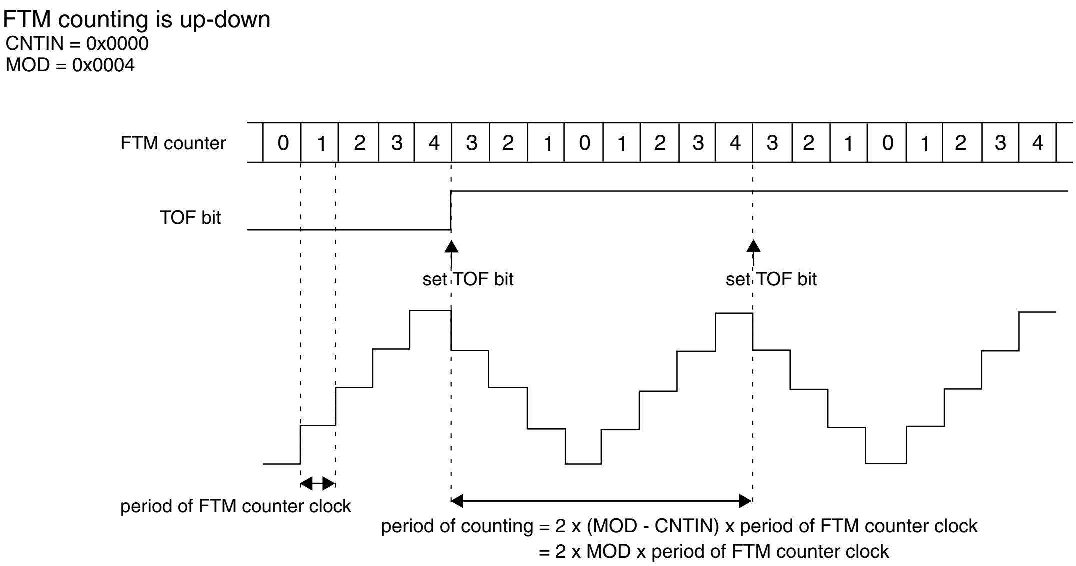


Figure 39-212. Example of up-down counting when CNTIN = 0x0000

Note

It is expected that the up-down counting be used only with CNTIN = 0x0000.

39.4.3.3 Free running counter

If (FTMEN = 0) and (MOD = 0x0000 or MOD = 0xFFFF), the FTM counter is a free running counter. In this case, the FTM counter runs free from 0x0000 through 0xFFFF and the TOF bit is set when the FTM counter changes from 0xFFFF to 0x0000. See the following figure.

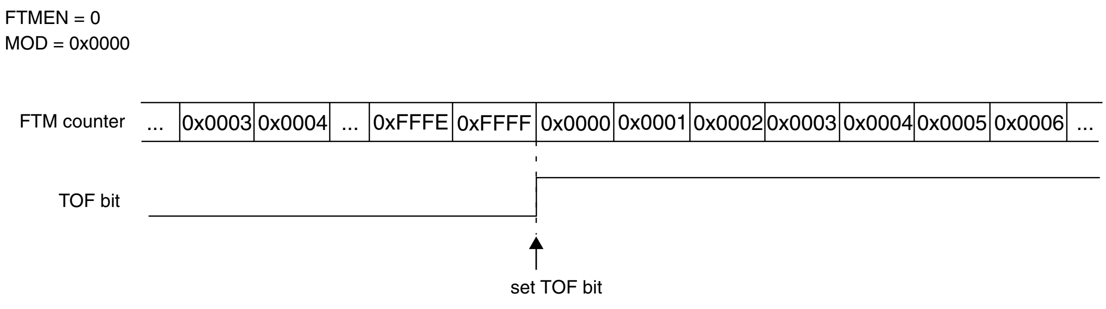


Figure 39-213. Example when the FTM counter is free running

The FTM counter is also a free running counter when:

- FTMEN = 1
- QUADEN = 0
- CPWMS = 0

- CNTIN = 0x0000, and
- MOD = 0xFFFF

39.4.3.4 Counter reset

Any one of the following cases resets the FTM counter to the value in the CNTIN register and the channels output to its initial value, except for channels in Output Compare mode.

- Any write to CNT.
- FTM counter synchronization.

39.4.3.5 When the TOF bit is set

The NUMTOF[4:0] bits define the number of times that the FTM counter overflow should occur before the TOF bit to be set. If NUMTOF[4:0] = 0x00, then the TOF bit is set at each FTM counter overflow.

Initialize the FTM counter, by writing to CNT, after writing to the NUMTOF[4:0] bits to avoid confusion about when the first counter overflow will occur.

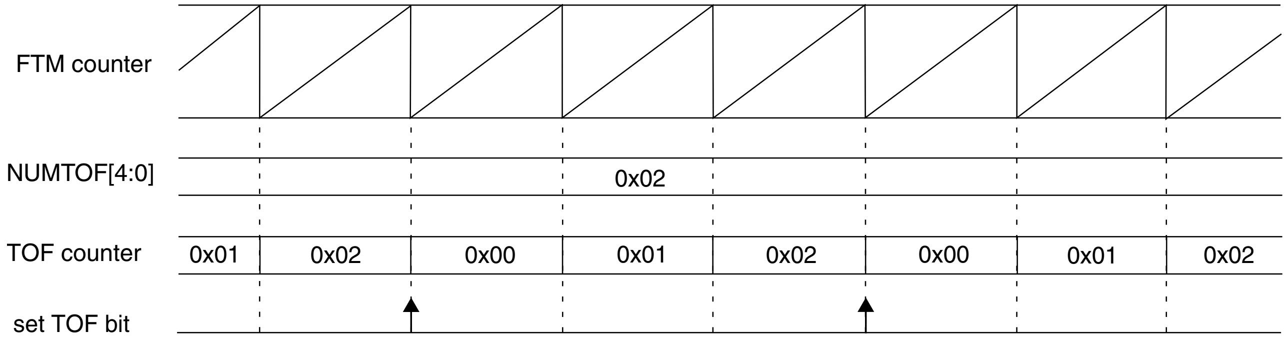


Figure 39-214. Periodic TOF when NUMTOF = 0x02

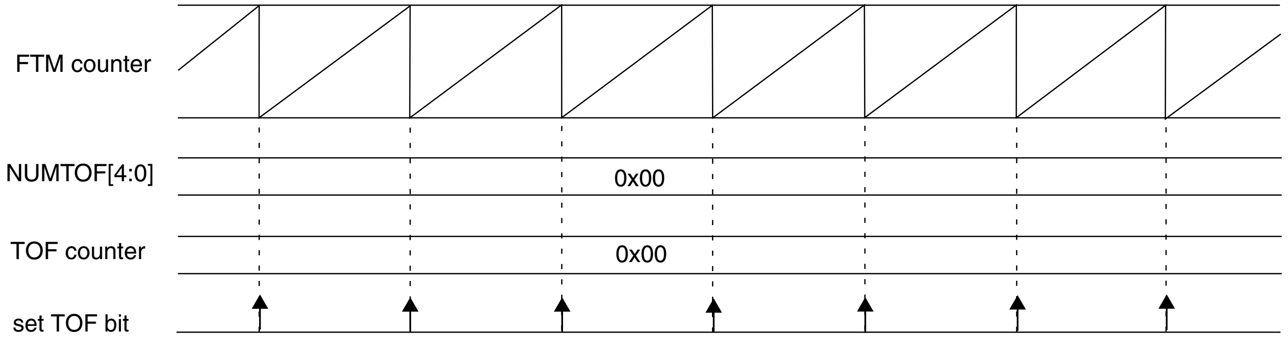


Figure 39-215. Periodic TOF when NUMTOF = 0x00

39.4.4 Input Capture mode

The Input Capture mode is selected when:

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0
- MSnB:MSnA = 0:0, and
- ELSnB:ELSnA ≠ 0:0

When a selected edge occurs on the channel input, the current value of the FTM counter is captured into the CnV register, at the same time the CHnF bit is set and the channel interrupt is generated if enabled by CHnIE = 1. See the following figure.

When a channel is configured for input capture, the FTMxCHn pin is an edge-sensitive input. ELSnB:ELSnA control bits determine which edge, falling or rising, triggers input-capture event. Note that the maximum frequency for the channel input signal to be detected correctly is system clock divided by 4, which is required to meet Nyquist criteria for signal sampling.

Writes to the CnV register is ignored in Input Capture mode.

While in BDM, the input capture function works as configured. When a selected edge event occurs, the FTM counter value, which is frozen because of BDM, is captured into the CnV register and the CHnF bit is set.

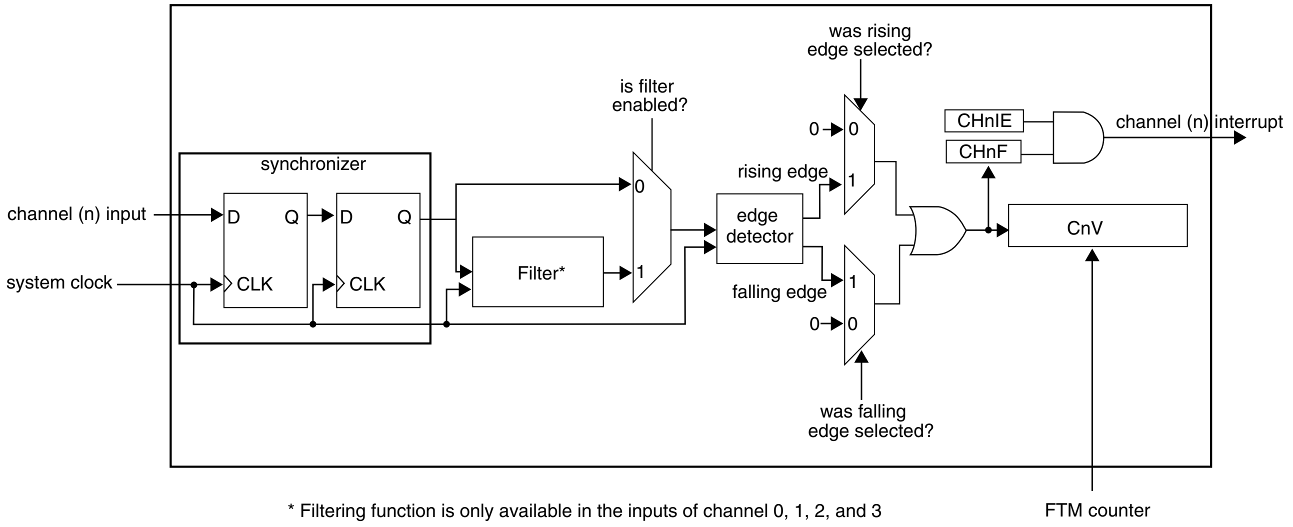


Figure 39-216. Input Capture mode

If the channel input does not have a filter enabled, then the input signal is always delayed 3 rising edges of the system clock, that is, two rising edges to the synchronizer plus one more rising edge to the edge detector. In other words, the CHnF bit is set on the third rising edge of the system clock after a valid edge occurs on the channel input.

Note

The Input Capture mode must be used only with CNTIN = 0x0000.

39.4.4.1 Filter for Input Capture mode

The filter function is only available on channels 0, 1, 2, and 3.

First, the input signal is synchronized by the system clock. Following synchronization, the input signal enters the filter block. See the following figure.

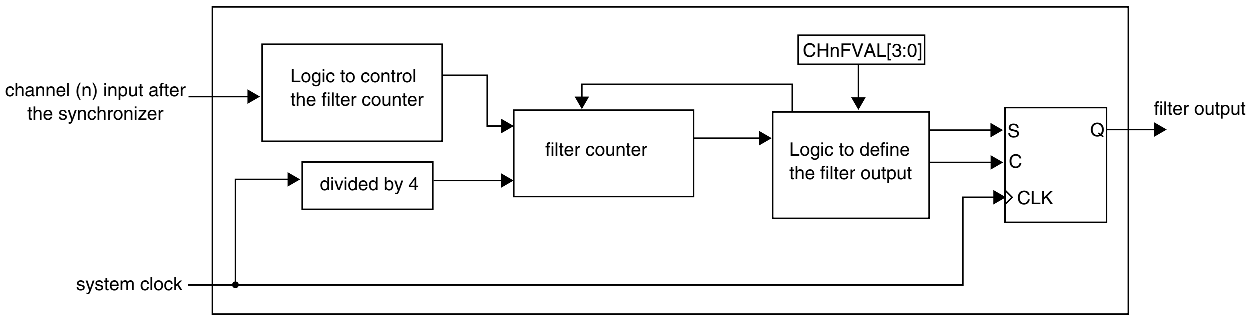


Figure 39-217. Channel input filter

When there is a state change in the input signal, the counter is reset and starts counting up. As long as the new state is stable on the input, the counter continues to increment. When the counter is equal to CHnFVAL[3:0], the state change of the input signal is validated. It is then transmitted as a pulse edge to the edge detector.

If the opposite edge appears on the input signal before it can be validated, the counter is reset. At the next input transition, the counter starts counting again. Any pulse that is shorter than the minimum value selected by CHnFVAL[3:0] (× 4 system clocks) is regarded as a glitch and is not passed on to the edge detector. A timing diagram of the input filter is shown in the following figure.

The filter function is disabled when CHnFVAL[3:0] bits are zero. In this case, the input signal is delayed 3 rising edges of the system clock. If (CHnFVAL[3:0] ≠ 0000), then the input signal is delayed by the minimum pulse width (CHnFVAL[3:0] × 4 system clocks) plus a further 4 rising edges of the system clock: two rising edges to the synchronizer,

one rising edge to the filter output, plus one more to the edge detector. In other words, CHnF is set (4 + 4 × CHnFVAL[3:0]) system clock periods after a valid edge occurs on the channel input.

The clock for the counter in the channel input filter is the system clock divided by 4.

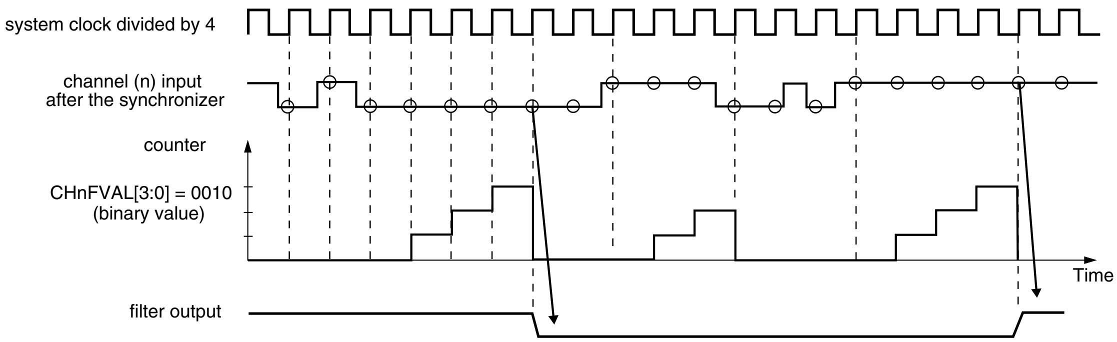


Figure 39-218. Channel input filter example

39.4.5 Output Compare mode

The Output Compare mode is selected when:

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB:MSnA = 0:1

In Output Compare mode, the FTM can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter matches the value in the CnV register of an output compare channel, the channel (n) output can be set, cleared, or toggled.

When a channel is initially configured to Toggle mode, the previous value of the channel output is held until the first output compare event occurs.

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM counter = CnV).

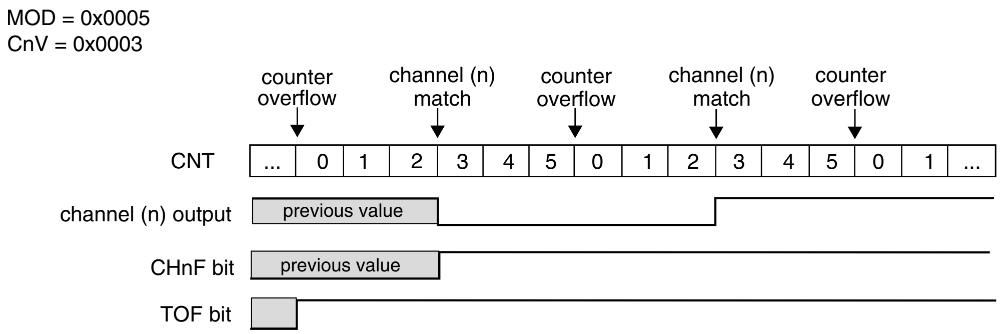


Figure 39-219. Example of the Output Compare mode when the match toggles the channel output

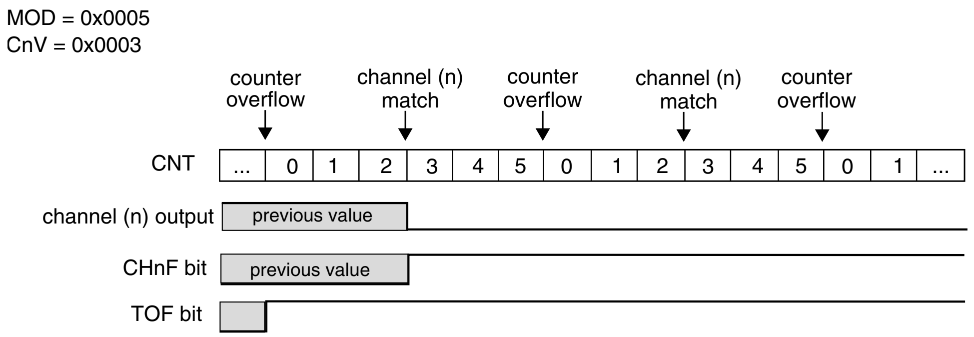


Figure 39-220. Example of the Output Compare mode when the match clears the channel output

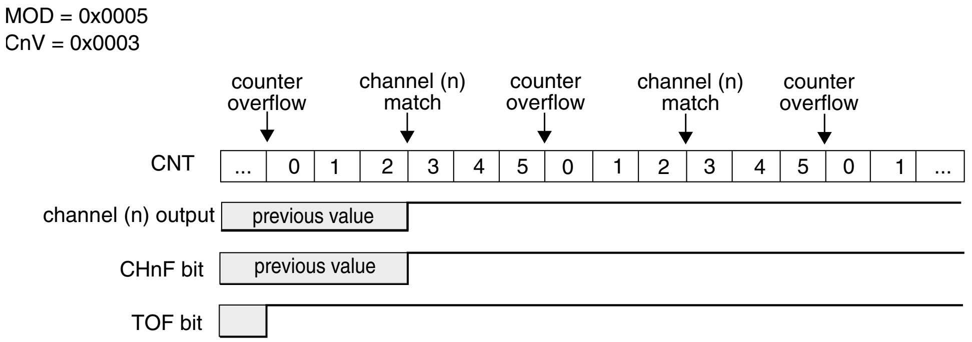


Figure 39-221. Example of the Output Compare mode when the match sets the channel output

If (ELSnB:ELSnA = 0:0) when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1, however the channel (n) output is not modified and controlled by FTM.

Note

The Output Compare mode must be used only with CNTIN = 0x0000.

39.4.6 Edge-Aligned PWM (EPWM) mode

The Edge-Aligned mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB = 1

The EPWM period is determined by (MOD − CNTIN + 0x0001) and the pulse width (duty cycle) is determined by (CnV − CNTIN).

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM counter = CnV), that is, at the end of the pulse width.

This type of PWM signal is called edge-aligned because the leading edges of all PWM signals are aligned with the beginning of the period, which is the same for all channels within an FTM.

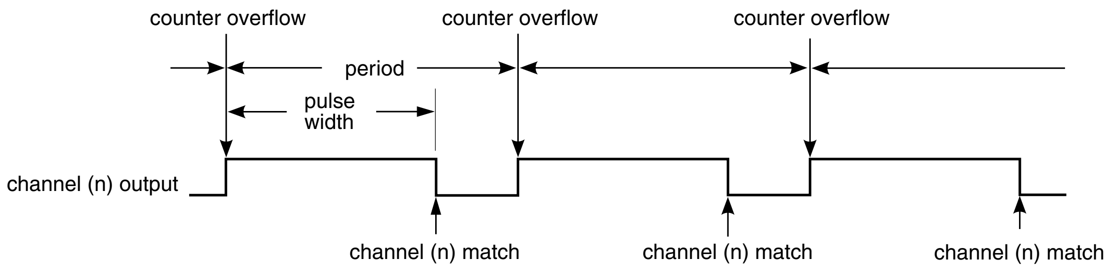


Figure 39-222. EPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1, however the channel (n) output is not controlled by FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced low at the channel (n) match (FTM counter = CnV). See the following figure.

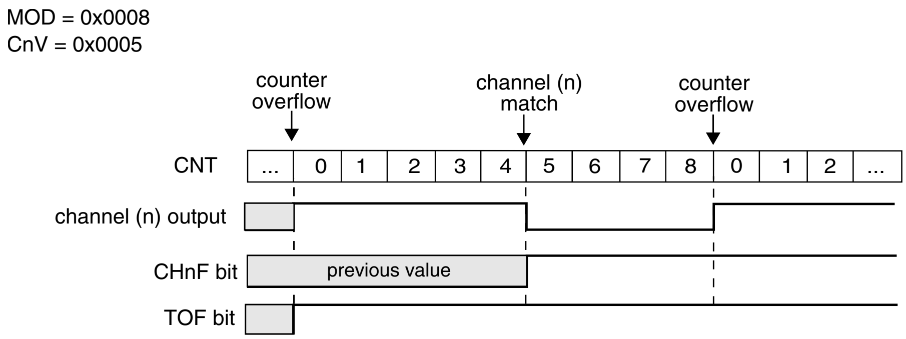


Figure 39-223. EPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced high at the channel (n) match (FTM counter = CnV). See the following figure.

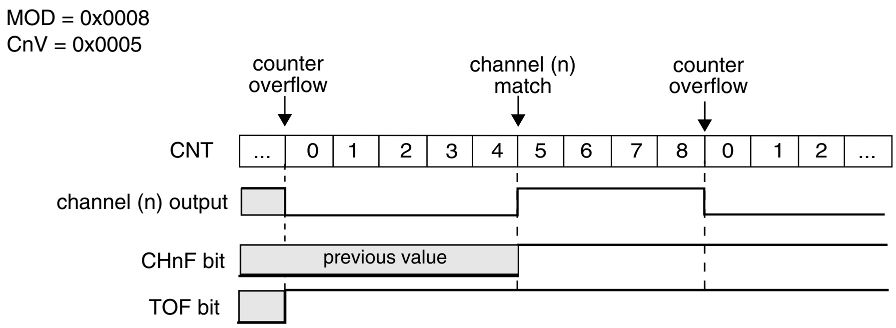


Figure 39-224. EPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000), then the channel (n) output is a 0% duty cycle EPWM signal and CHnF bit is not set even when there is the channel (n) match. If (CnV > MOD), then the channel (n) output is a 100% duty cycle EPWM signal and CHnF bit is not set even when there is the channel (n) match. Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle EPWM signal.

Note

The EPWM mode must be used only with CNTIN = 0x0000.

39.4.7 Center-Aligned PWM (CPWM) mode

The Center-Aligned mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMBINE = 0, and
- CPWMS = 1

The CPWM pulse width (duty cycle) is determined by 2 × (CnV − CNTIN) and the period is determined by 2 × (MOD − CNTIN). See the following figure. MOD must be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results.

In the CPWM mode, the FTM counter counts up until it reaches MOD and then counts down until it reaches CNTIN.

The CHnF bit is set and channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM counter = CnV) when the FTM counting is down (at the begin of the pulse width) and when the FTM counting is up (at the end of the pulse width).

This type of PWM signal is called center-aligned because the pulse width centers for all channels are aligned with the value of CNTIN.

The other channel modes are not compatible with the up-down counter (CPWMS = 1). Therefore, all FTM channels must be used in CPWM mode when (CPWMS = 1).

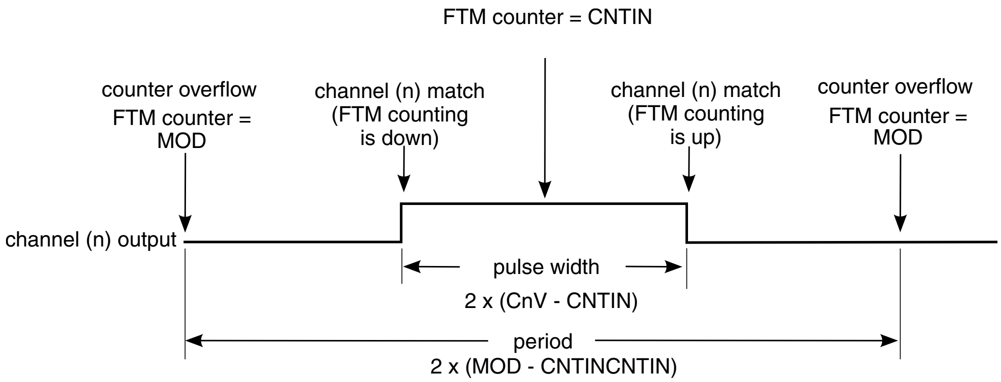


Figure 39-225. CPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the FTM counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not controlled by FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the channel (n) match (FTM counter = CnV) when counting down, and it is forced low at the channel (n) match when counting up. See the following figure.

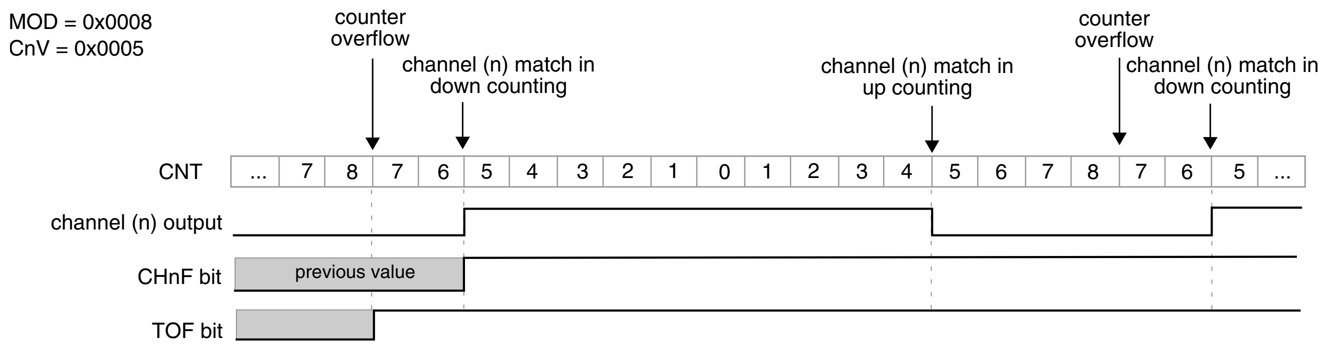


Figure 39-226. CPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the channel (n) match (FTM counter = CnV) when counting down, and it is forced high at the channel (n) match when counting up. See the following figure.

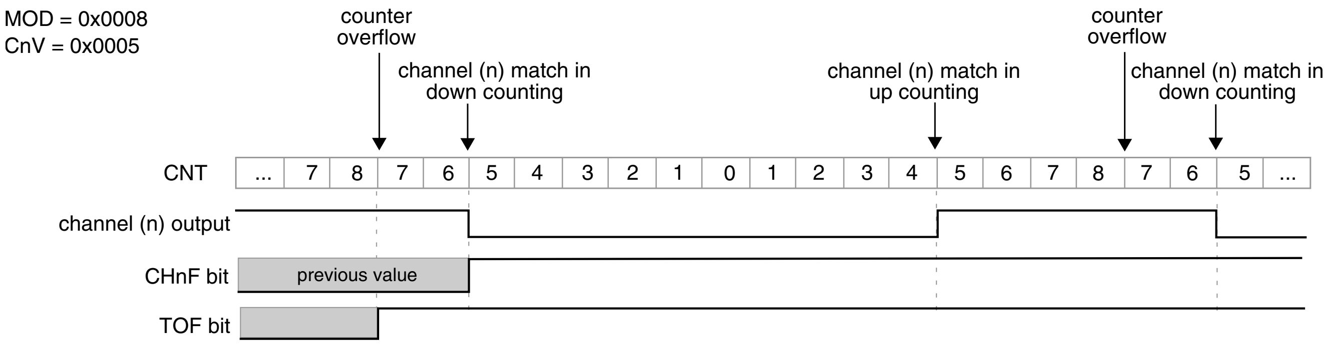


Figure 39-227. CPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000) or CnV is a negative value, that is (CnV[15] = 1), then the channel (n) output is a 0% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match.

If CnV is a positive value, that is (CnV[15] = 0), (CnV ≥ MOD), and (MOD ≠ 0x0000), then the channel (n) output is a 100% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match. This implies that the usable range of periods set by MOD is 0x0001 through 0x7FFE, 0x7FFF if you do not need to generate a 100% duty cycle CPWM signal. This is not a significant limitation because the resulting period is much longer than required for normal applications.

The CPWM mode must not be used when the FTM counter is a free running counter.

Note

The CPWM mode must be used only with CNTIN = 0x0000.

39.4.8 Combine mode

The Combine mode is selected when:

- FTMEN = 1
- QUADEN = 0
- DECAPEN = 0
- COMBINE = 1, and
- CPWMS = 0

In Combine mode, an even channel (n) and adjacent odd channel (n+1) are combined to generate a PWM signal in the channel (n) output.

In the Combine mode, the PWM period is determined by (MOD − CNTIN + 0x0001) and the PWM pulse width (duty cycle) is determined by (|C(n+1)V − C(n)V|).

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM counter = C(n)V). The CH(n+1)F bit is set and the channel (n +1) interrupt is generated, if CH(n+1)IE = 1, at the channel (n+1) match (FTM counter = C(n+1)V).

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced high at the channel (n) match (FTM counter = C(n)V). See the following figure.

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced high at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced low at the channel (n) match (FTM counter = C(n)V). See the following figure.

In Combine mode, the ELS(n+1)B and ELS(n+1)A bits are not used in the generation of the channels (n) and (n+1) output. However, if (ELSnB:ELSnA = 0:0) then the channel (n) output is not controlled by FTM, and if (ELS(n+1)B:ELS(n+1)A = 0:0) then the channel (n+1) output is not controlled by FTM.

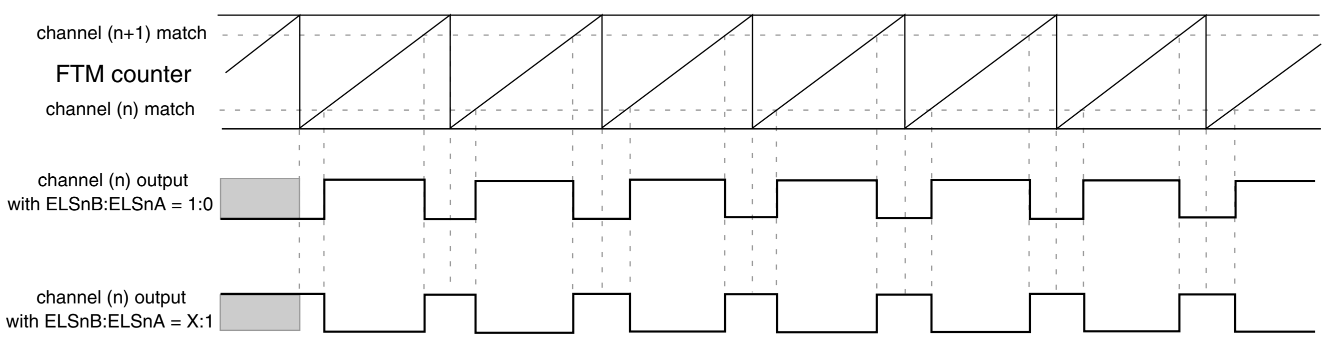


Figure 39-228. Combine mode

The following figures illustrate the PWM signals generation using Combine mode.

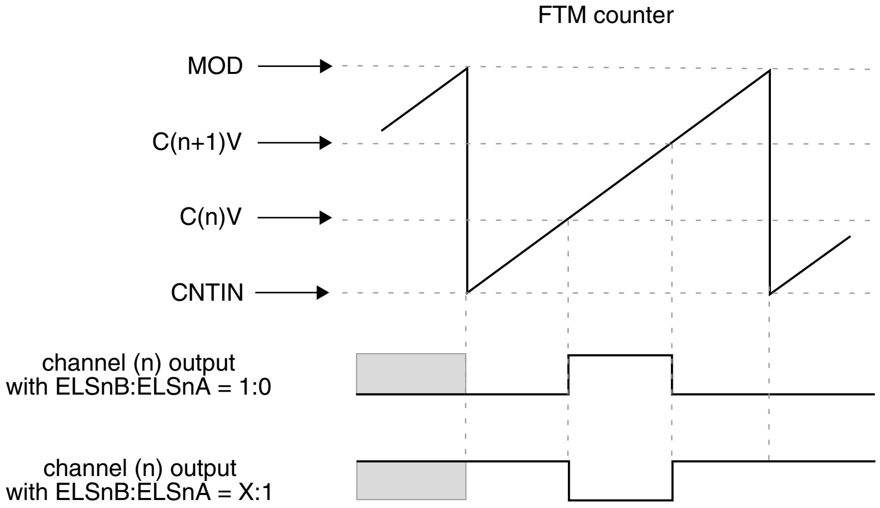


Figure 39-229. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V < C(n+1)V)

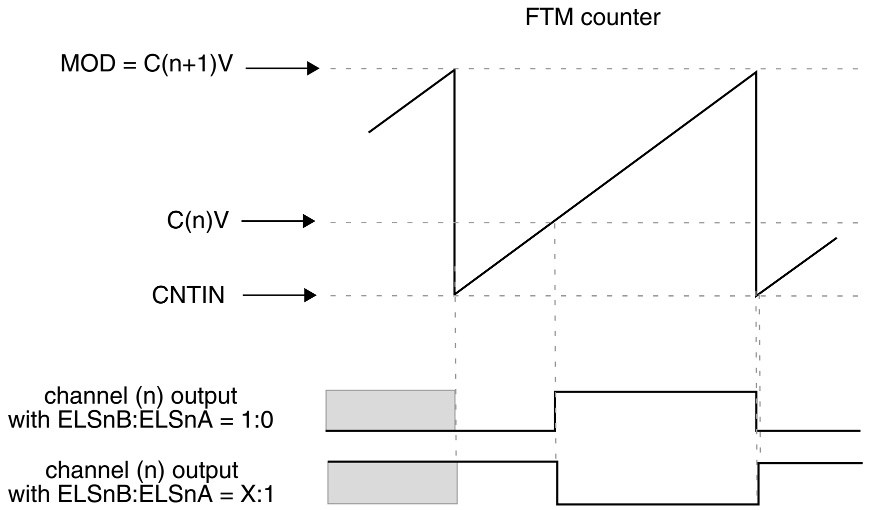


Figure 39-230. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n+1)V = MOD)

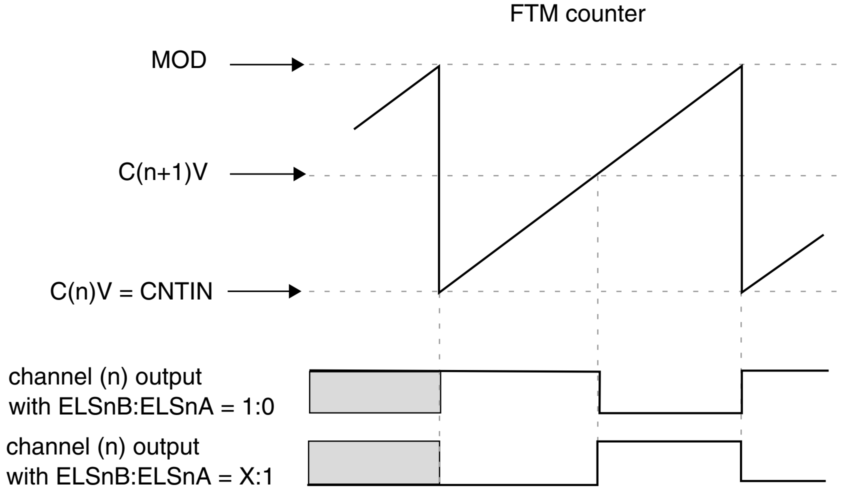


Figure 39-231. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD)

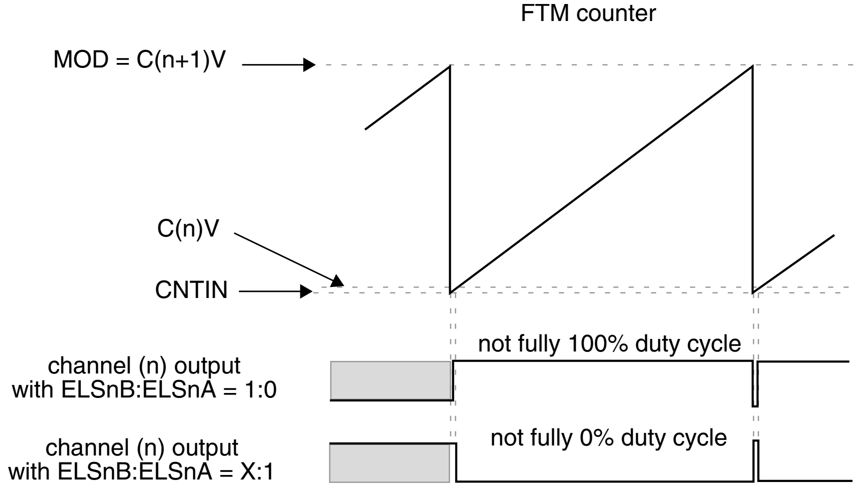


Figure 39-232. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n)V is Almost Equal to CNTIN) and (C(n+1)V = MOD)

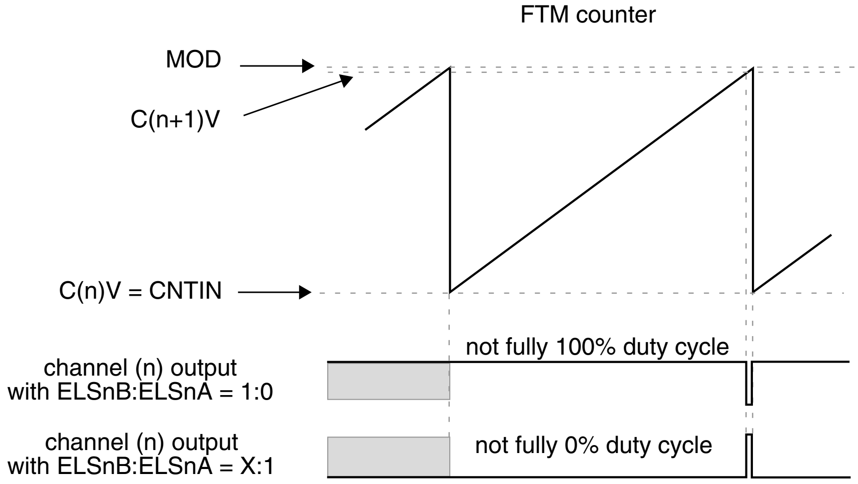


Figure 39-233. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD) and (C(n+1)V is Almost Equal to MOD)

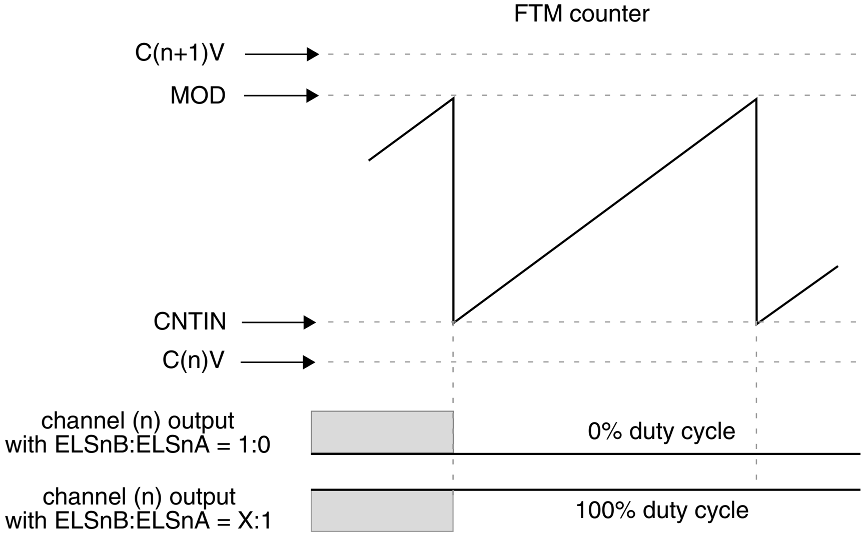


Figure 39-234. Channel (n) output if C(n)V and C(n+1)V are not between CNTIN and MOD

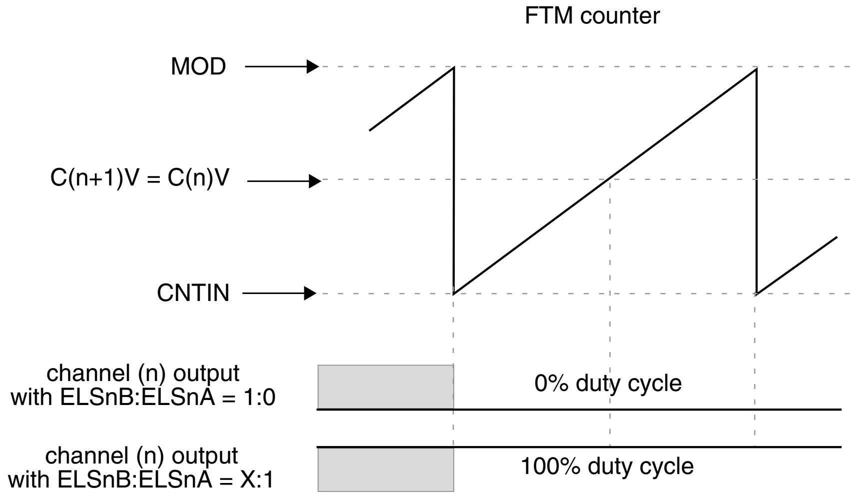


Figure 39-235. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V = C(n+1)V)

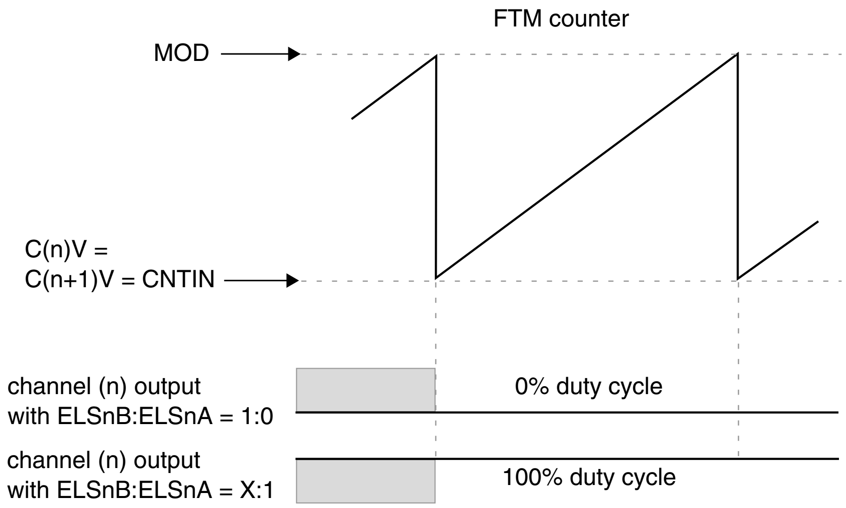


Figure 39-236. Channel (n) output if (C(n)V = C(n+1)V = CNTIN)

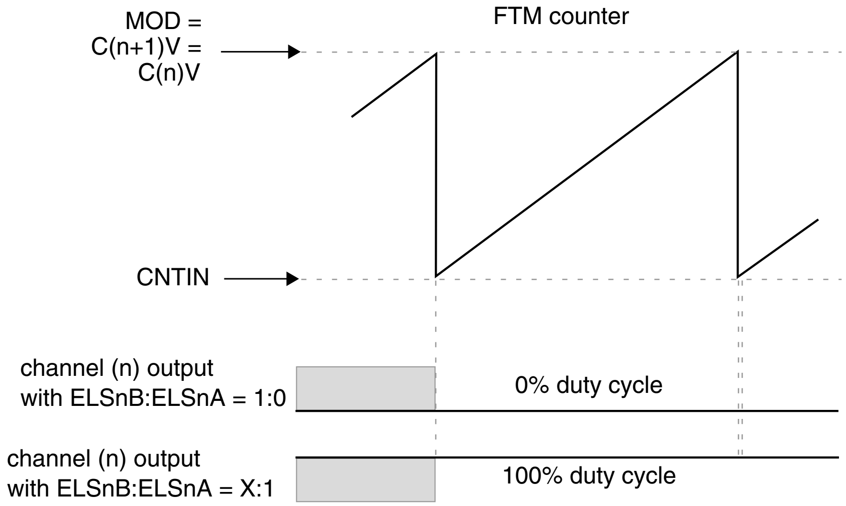


Figure 39-237. Channel (n) output if (C(n)V = C(n+1)V = MOD)

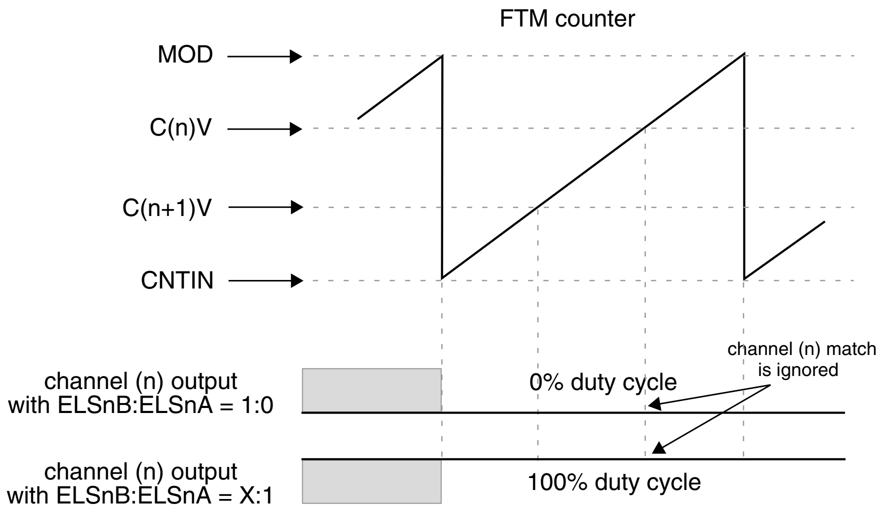


Figure 39-238. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V > C(n+1)V)

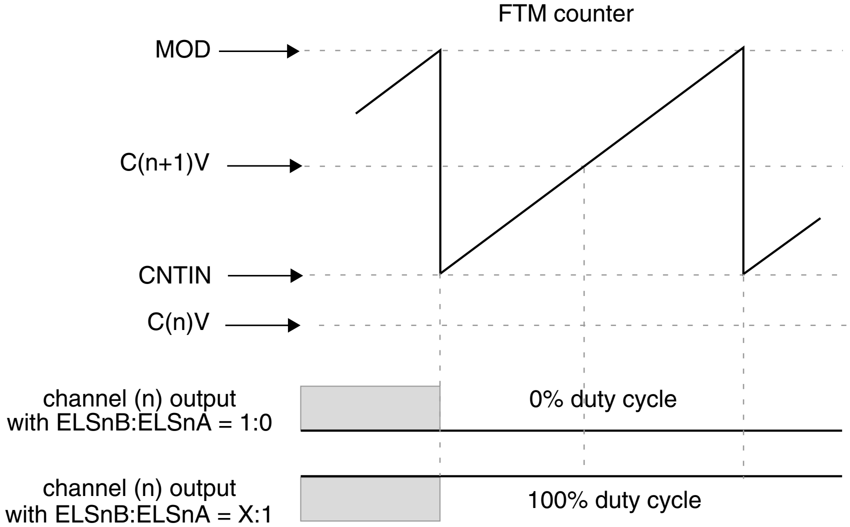


Figure 39-239. Channel (n) output if (C(n)V < CNTIN) and (CNTIN < C(n+1)V < MOD)

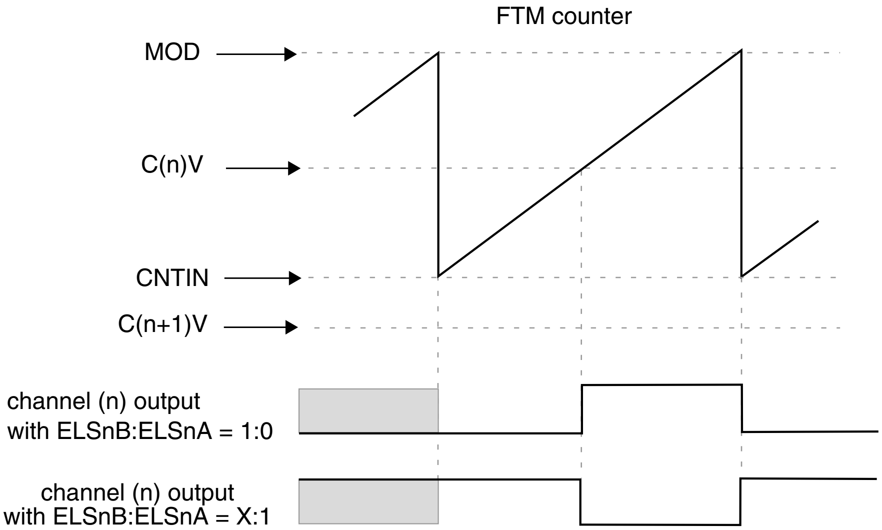


Figure 39-240. Channel (n) output if (C(n+1)V < CNTIN) and (CNTIN < C(n)V < MOD)

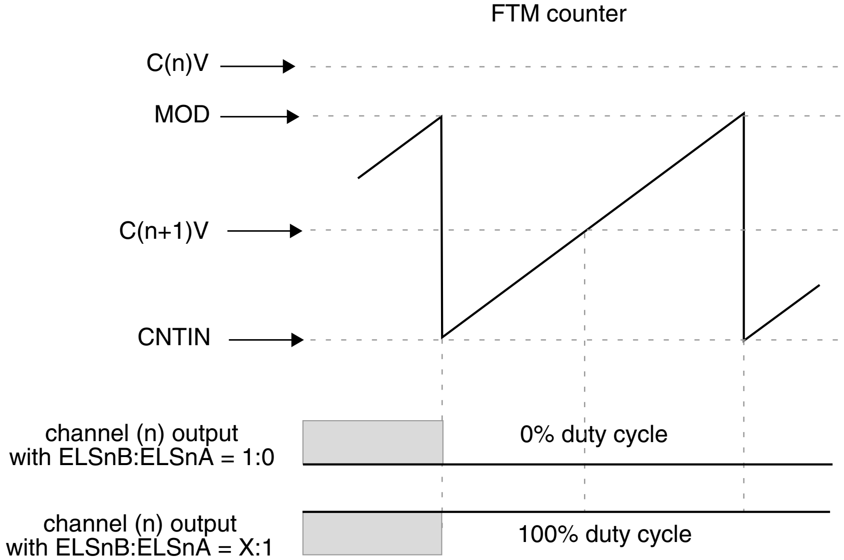


Figure 39-241. Channel (n) output if (C(n)V > MOD) and (CNTIN < C(n+1)V < MOD)

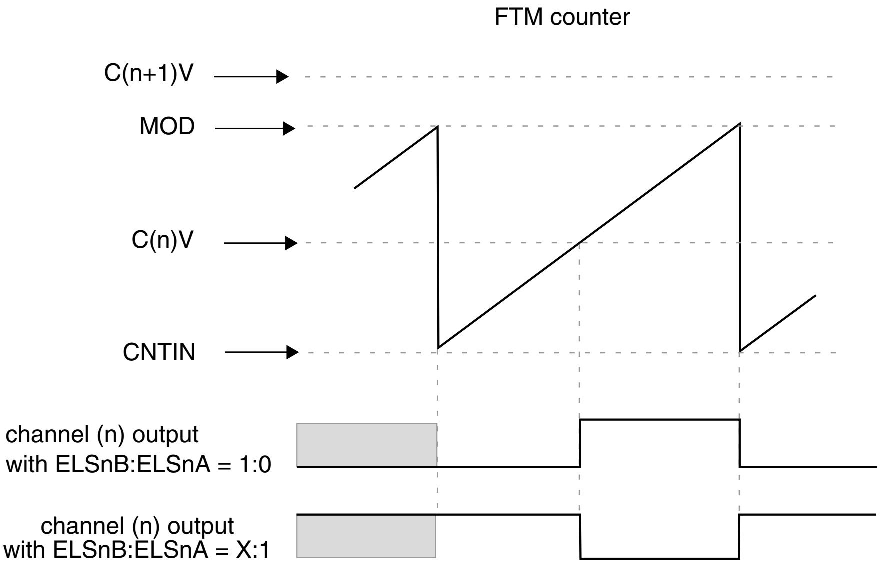


Figure 39-242. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V < MOD)

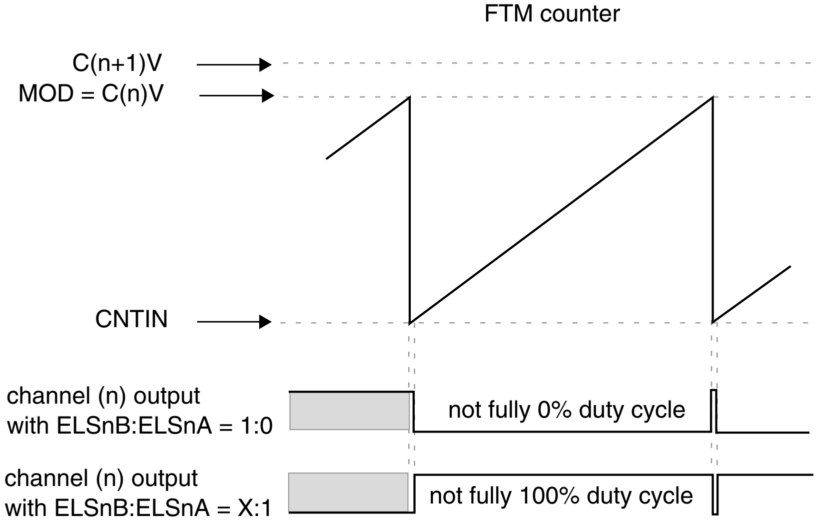


Figure 39-243. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V = MOD)

39.4.8.1 Asymmetrical PWM

In Combine mode, the control of the PWM signal first edge, when the channel (n) match occurs, that is, FTM counter = C(n)V, is independent of the control of the PWM signal second edge, when the channel (n+1) match occurs, that is, FTM counter = C(n+1)V. So, Combine mode allows the generation of asymmetrical PWM signals.

39.4.9 Complementary mode

The Complementary mode is selected when:

- FTMEN = 1
- QUADEN = 0
- DECAPEN = 0
- COMBINE = 1
- CPWMS = 0, and
- COMP = 1

In Complementary mode, the channel (n+1) output is the inverse of the channel (n) output.

So, the channel (n+1) output is the same as the channel (n) output when:

- FTMEN = 1
- QUADEN = 0
- DECAPEN = 0
- COMBINE = 1

- CPWMS = 0, and
- COMP = 0

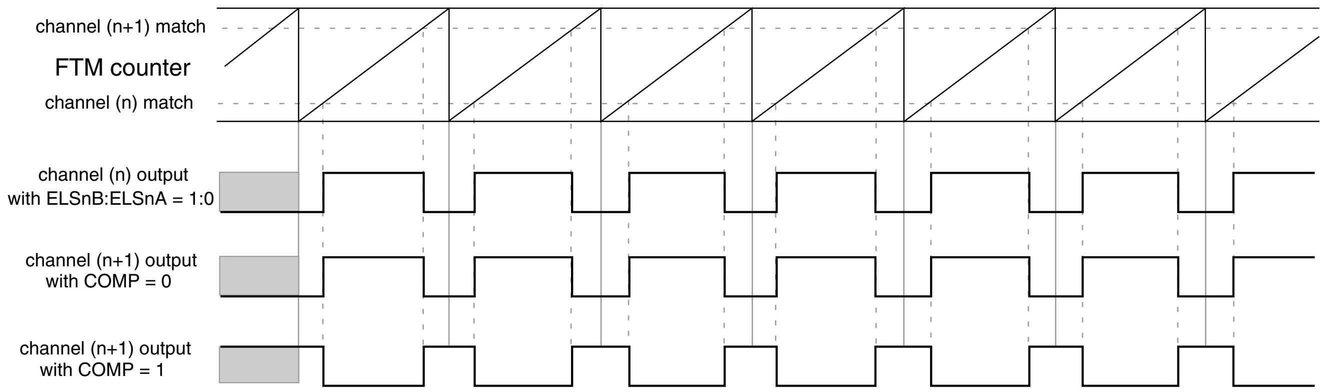


Figure 39-244. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = 1:0)

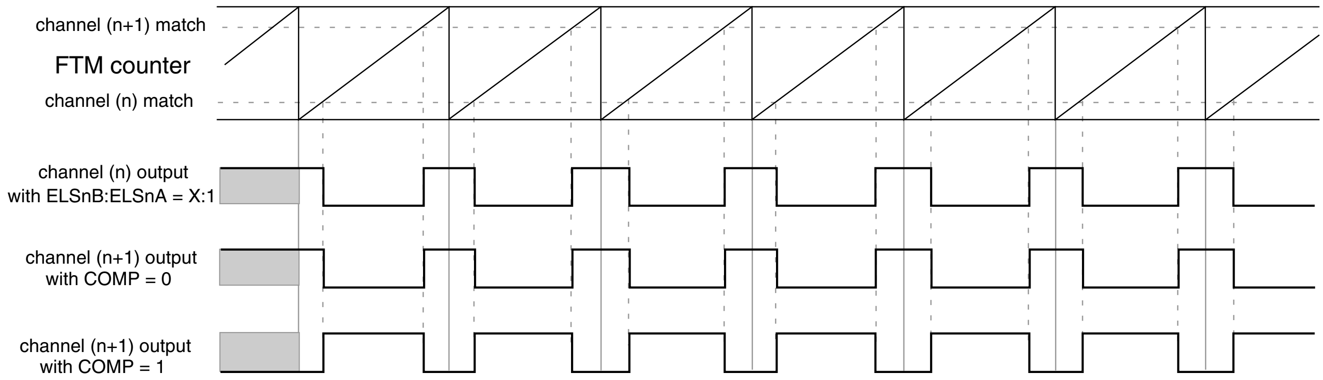


Figure 39-245. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = X:1)

39.4.10 Registers updated from write buffers

39.4.10.1 CNTIN register update

The following table describes when CNTIN register is updated:

Table 39-303. CNTIN register update

When	Then CNTIN register is updated
CLKS[1:0] = 0:0	When CNTIN register is written, independent of FTMEN bit.
• FTMEN = 0, or • CNTINC = 0	At the next system clock after CNTIN was written.
• FTMEN = 1, • SYNCMODE = 1, and • CNTINC = 1	By the CNTIN register synchronization.

39.4.10.2 MOD register update

The following table describes when MOD register is updated:

Table 39-304. MOD register update

When	Then MOD register is updated
CLKS[1:0] = 0:0	When MOD register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the CPWMS bit, that is: • If the selected mode is not CPWM then MOD register is updated after MOD register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM then MOD register is updated after MOD register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	By the MOD register synchronization.

39.4.10.3 CnV register update

The following table describes when CnV register is updated:

Table 39-305. CnV register update

When	Then CnV register is updated
CLKS[1:0] = 0:0	When CnV register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the selected mode, that is: • If the selected mode is Output Compare, then CnV register is updated on the next FTM counter change, end of the prescaler counting, after CnV register was written. • If the selected mode is EPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	According to the selected mode, that is: • If the selected mode is output compare then CnV register is updated according to the SYNCEN bit. If (SYNCEN = 0) then CnV register is updated after CnV register was written at the next change of the FTM counter, the end of the prescaler counting. If (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization. • If the selected mode is not output compare and (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization.

39.4.11 PWM synchronization

The PWM synchronization provides an opportunity to update the MOD, CNTIN, CnV, OUTMASK, INVCTRL and SWOCTRL registers with their buffered value and force the FTM counter to the CNTIN register value.

Note

- The PWM synchronization must be used only in Combine mode.
- The legacy PWM synchronization (SYNCMODE = 0) is a subset of the enhanced PWM synchronization (SYNCMODE = 1). Thus, only the enhanced PWM synchronization must be used.

39.4.11.1 Hardware trigger

Three hardware trigger signal inputs of the FTM module are enabled when TRIGn = 1, where n = 0, 1 or 2 corresponding to each one of the input signals, respectively. The hardware trigger input n is synchronized by the system clock. The PWM synchronization with hardware trigger is initiated when a rising edge is detected at the enabled hardware trigger inputs.

If (HWTRIGMODE = 0) then the TRIGn bit is cleared when 0 is written to it or when the trigger n event is detected.

In this case, if two or more hardware triggers are enabled (for example, TRIG0 and TRIG1 = 1) and only trigger 1 event occurs, then only TRIG1 bit is cleared. If a trigger n event occurs together with a write setting TRIGn bit, then the synchronization is initiated, but TRIGn bit remains set due to the write operation.

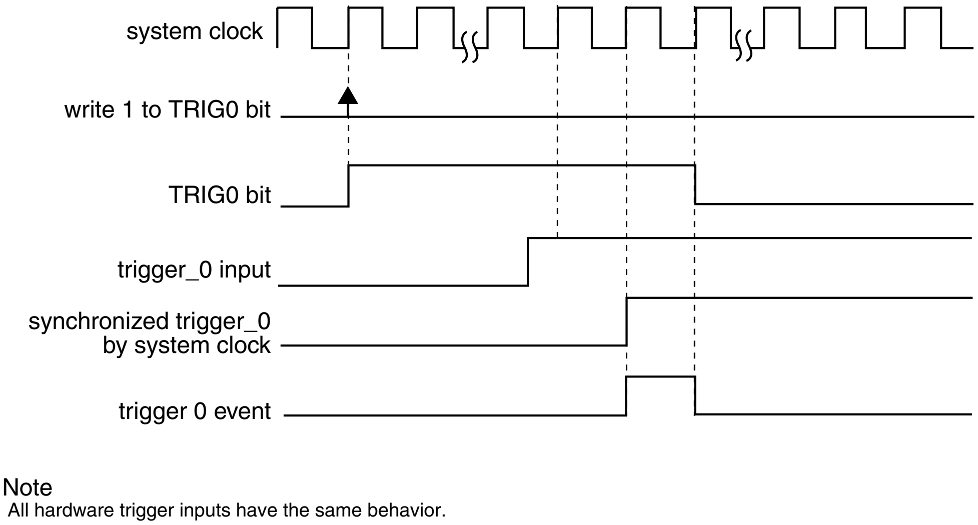


Figure 39-246. Hardware trigger event with HWTRIGMODE = 0

If HWTRIGMODE = 1, then the TRIGn bit is only cleared when 0 is written to it.

NOTE

The HWTRIGMODE bit must be 1 only with enhanced PWM synchronization (SYNCMODE = 1).

39.4.11.2 Software trigger

A software trigger event occurs when 1 is written to the SYNC[SWSYNC] bit. The SWSYNC bit is cleared when 0 is written to it or when the PWM synchronization, initiated by the software event, is completed.

If another software trigger event occurs (by writing another 1 to the SWSYNC bit) at the same time the PWM synchronization initiated by the previous software trigger event is ending, a new PWM synchronization is started and the SWSYNC bit remains equal to 1.

If SYNCMODE = 0 then the SWSYNC bit is also cleared by FTM according to PWMSYNC and REINIT bits. In this case if (PWMSYNC = 1) or (PWMSYNC = 0 and REINIT = 0) then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see Boundary cycle and loading points and the following figure. If (PWMSYNC = 0) and (REINIT = 1) then SWSYNC bit is cleared when the software trigger event occurs.

If SYNCMODE = 1 then the SWSYNC bit is also cleared by FTM according to the SWRSTCNT bit. If SWRSTCNT = 0 then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see the following figure. If SWRSTCNT = 1 then SWSYNC bit is cleared when the software trigger event occurs.

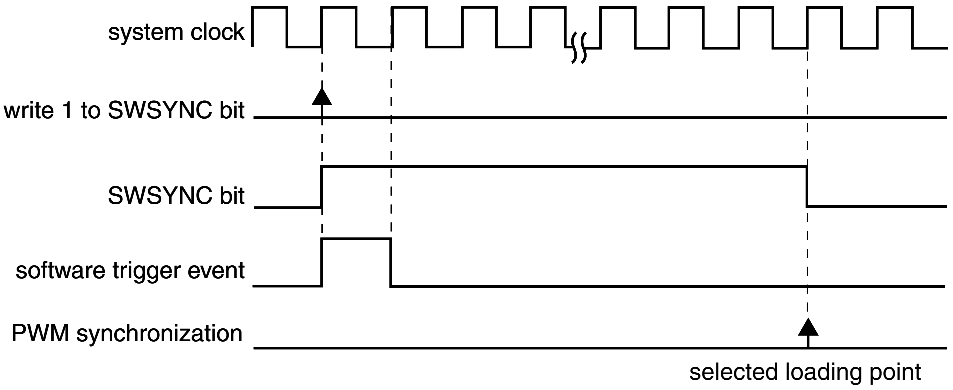


Figure 39-247. Software trigger event

39.4.11.3 Boundary cycle and loading points

The boundary cycle definition is important for the loading points for the registers MOD, CNTIN, and C(n)V.

In Up counting mode, the boundary cycle is defined as when the counter wraps to its initial value (CNTIN). If in Up-down counting mode, then the boundary cycle is defined as when the counter turns from down to up counting and when from up to down counting.

The following figure shows the boundary cycles and the loading points for the registers. In the Up Counting mode, the loading points are enabled if one of CNTMIN or CTMAX bits are 1. In the Up-Down Counting mode, the loading points are selected by CNTMIN and CNTMAX bits, as indicated in the figure. These loading points are safe places for register updates thus allowing a smooth transitions in PWM waveform generation.

For both counting modes, if neither CNTMIN nor CNTMAX are 1, then the boundary cycles are not used as loading points for registers updates. See the register synchronization descriptions in the following sections for details.

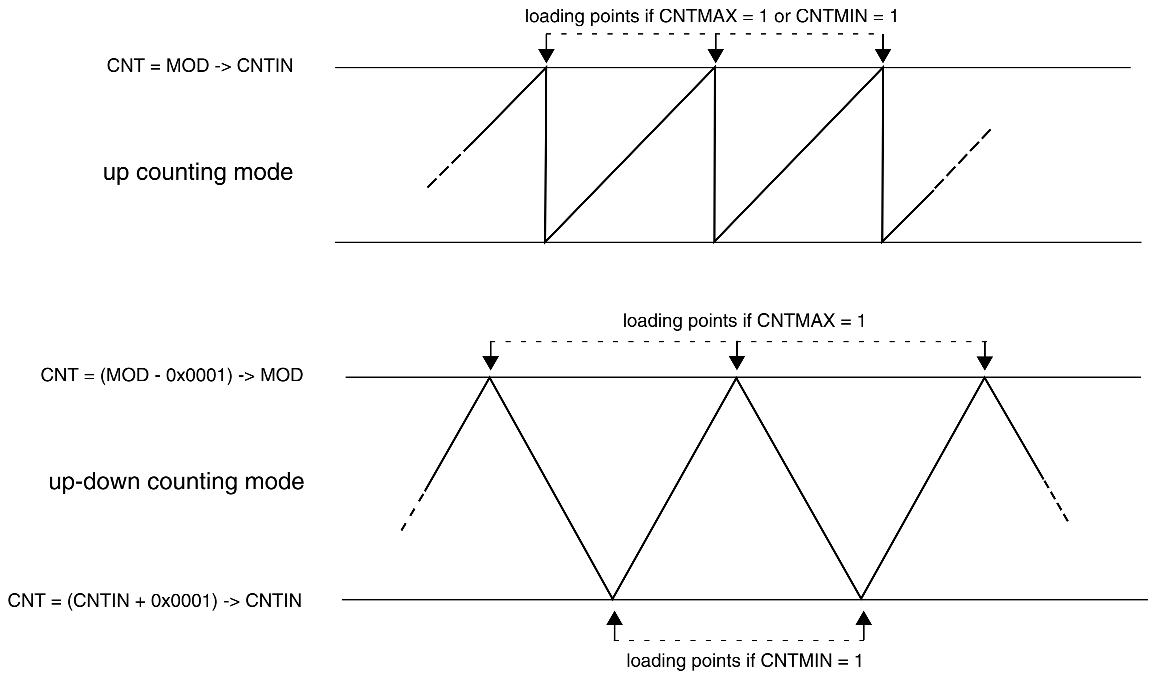


Figure 39-248. Boundary cycles and loading points

39.4.11.4 MOD register synchronization

The MOD register synchronization updates the MOD register with its buffer value. This synchronization is enabled if (FTMEN = 1).

The MOD register synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, it is expected that the MOD register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the MOD register synchronization depends on SWWRBUF, SWRSTCNT, HWWRBUF, and HWRSTCNT bits according to this flowchart:

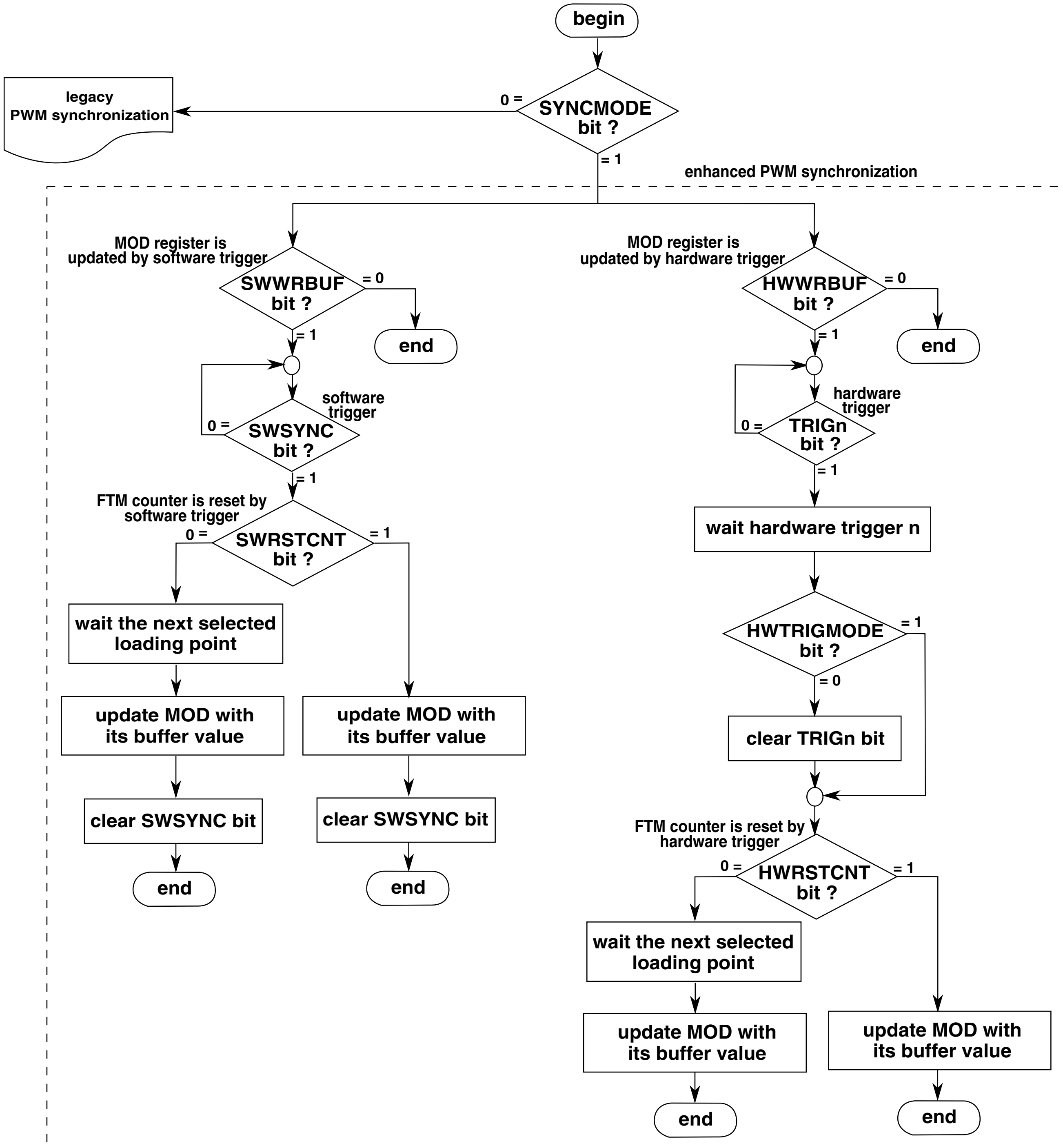


Figure 39-249. MOD register synchronization flowchart

In the case of legacy PWM synchronization, the MOD register synchronization depends on PWMSYNC and REINIT bits according to the following description.

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 0), then this synchronization is made on the next selected loading point after an enabled trigger event takes place. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected

loading point. If the trigger event was a hardware trigger, then the trigger enable bit (TRIGn) is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

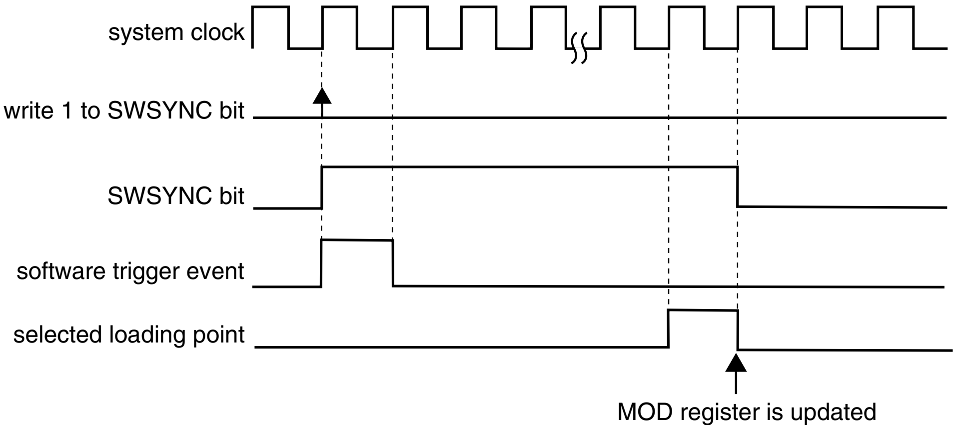


Figure 39-250. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 0), and software trigger was used

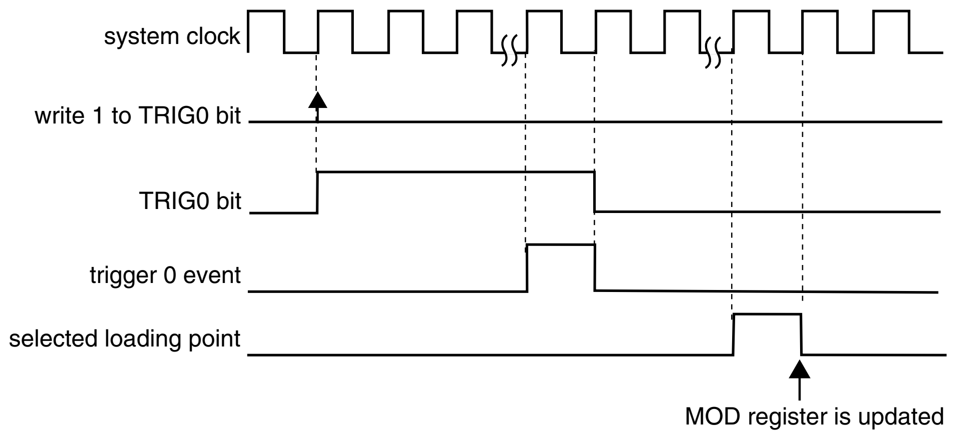


Figure 39-251. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 0), and a hardware trigger was used

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 1), then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

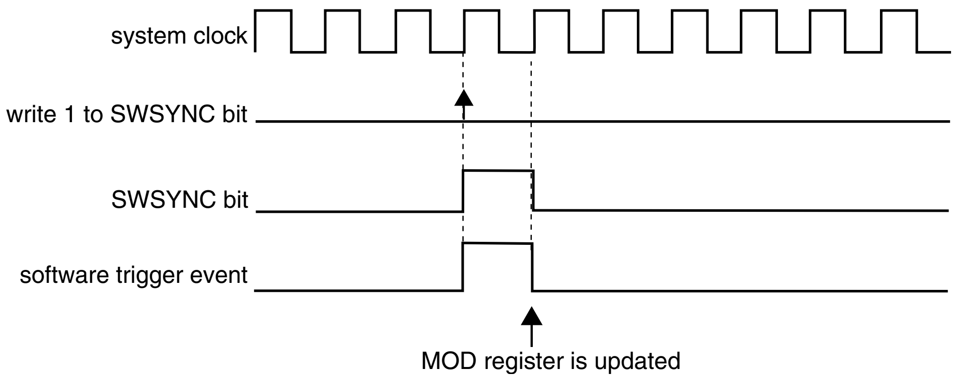


Figure 39-252. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 1), and software trigger was used

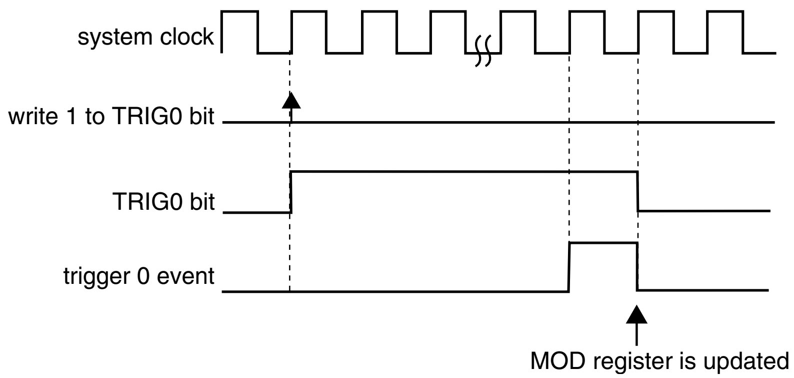


Figure 39-253. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 1), and a hardware trigger was used

If (SYNCMODE = 0) and (PWMSYNC = 1), then this synchronization is made on the next selected loading point after the software trigger event takes place. The SWSYNC bit is cleared on the next selected loading point:

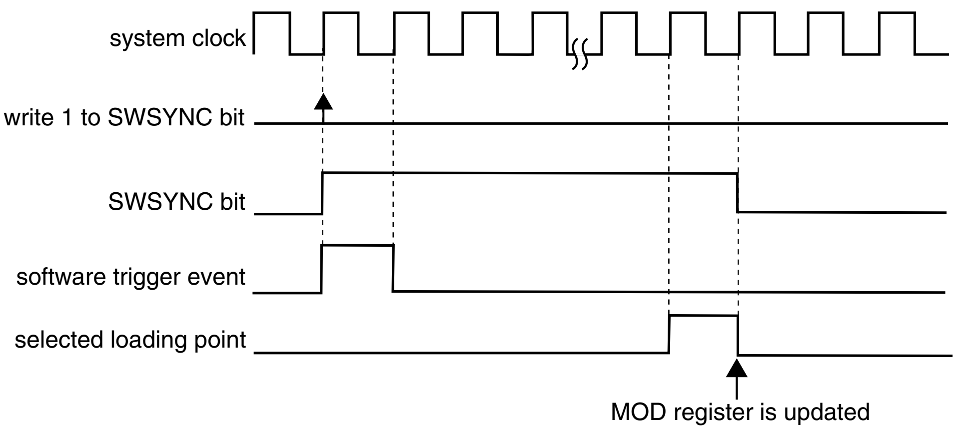


Figure 39-254. MOD synchronization with (SYNCMODE = 0) and (PWMSYNC = 1)

39.4.11.5 CNTIN register synchronization

The CNTIN register synchronization updates the CNTIN register with its buffer value.

This synchronization is enabled if (FTMEN = 1), (SYNCMODE = 1), and (CNTINC = 1). The CNTIN register synchronization can be done only by the enhanced PWM synchronization (SYNCMODE = 1). The synchronization mechanism is the same as the MOD register synchronization done by the enhanced PWM synchronization; see MOD register synchronization.

39.4.11.6 C(n)V and C(n+1)V register synchronization

The C(n)V and C(n+1)V registers synchronization updates the C(n)V and C(n+1)V registers with their buffer values.

This synchronization is enabled if (FTMEN = 1) and (SYNCEN = 1). The synchronization mechanism is the same as the MOD register synchronization. However, it is expected that the C(n)V and C(n+1)V registers be synchronized only by the enhanced PWM synchronization (SYNCMODE = 1).

39.4.11.7 OUTMASK register synchronization

The OUTMASK register synchronization updates the OUTMASK register with its buffer value.

The OUTMASK register can be updated at each rising edge of system clock (SYNCHOM = 0), by the enhanced PWM synchronization (SYNCHOM = 1 and SYNCMODE = 1) or by the legacy PWM synchronization (SYNCHOM = 1 and SYNCMODE = 0). However, it is expected that the OUTMASK register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the OUTMASK register synchronization depends on SWOM and HWOM bits. See the following flowchart:

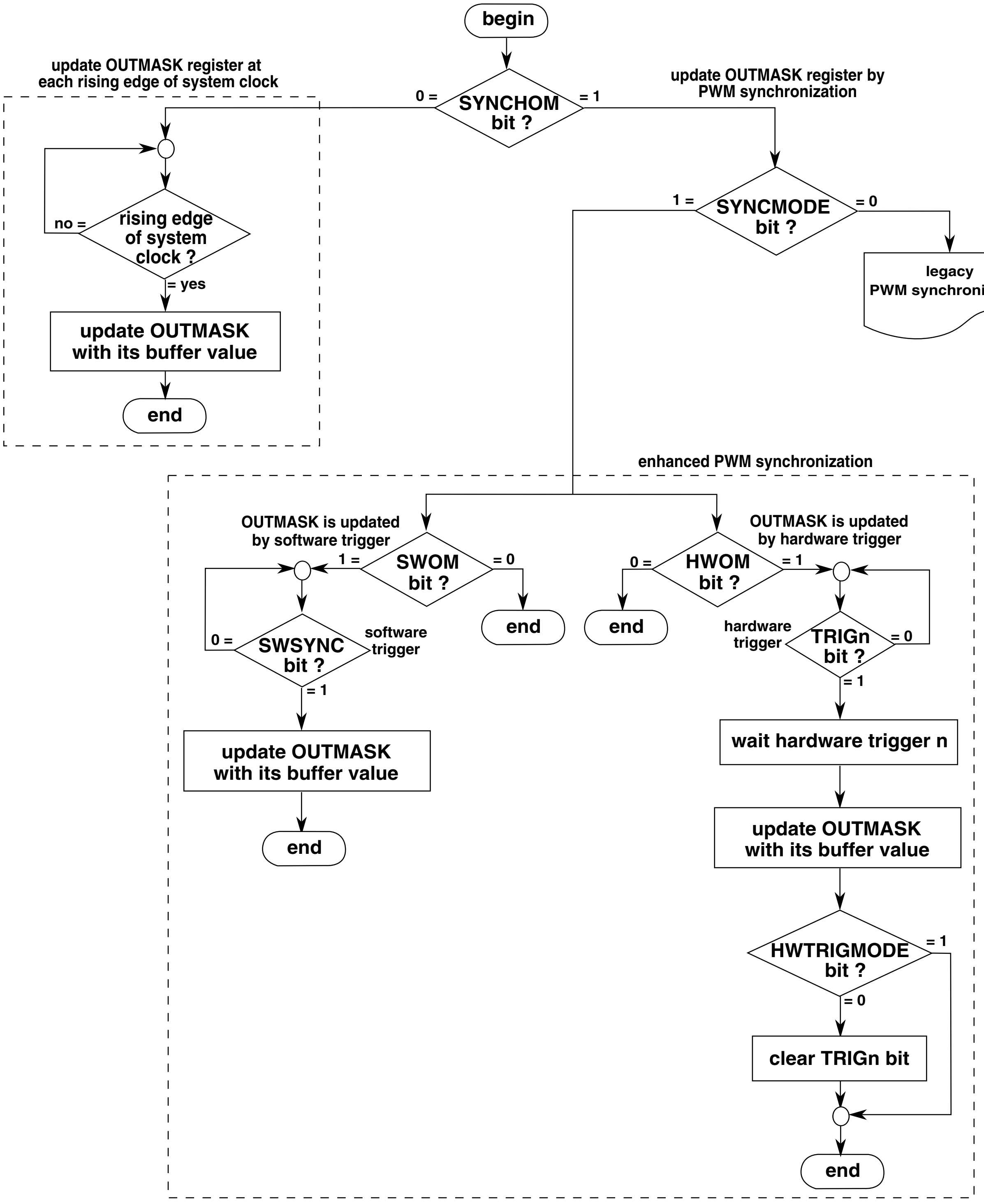


Figure 39-255. OUTMASK register synchronization flowchart

In the case of legacy PWM synchronization, the OUTMASK register synchronization depends on PWMSYNC bit according to the following description.

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 0), then this synchronization is done on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected loading point. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

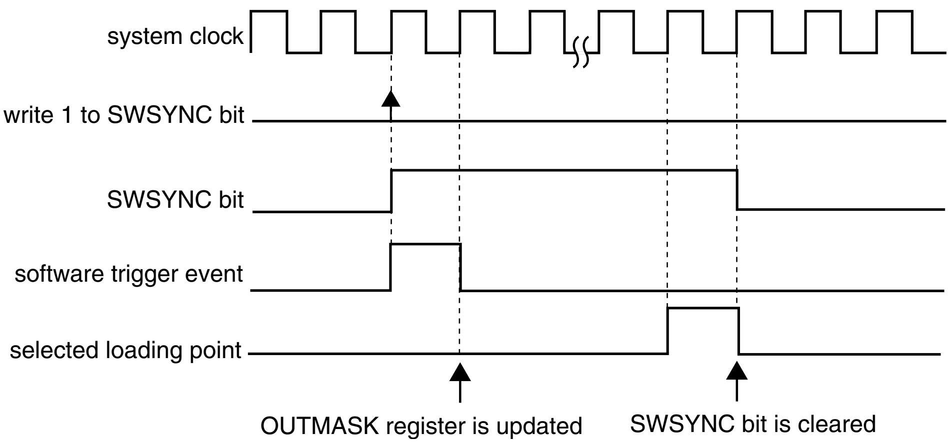


Figure 39-256. OUTMASK synchronization with (SYNCMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0) and software trigger was used

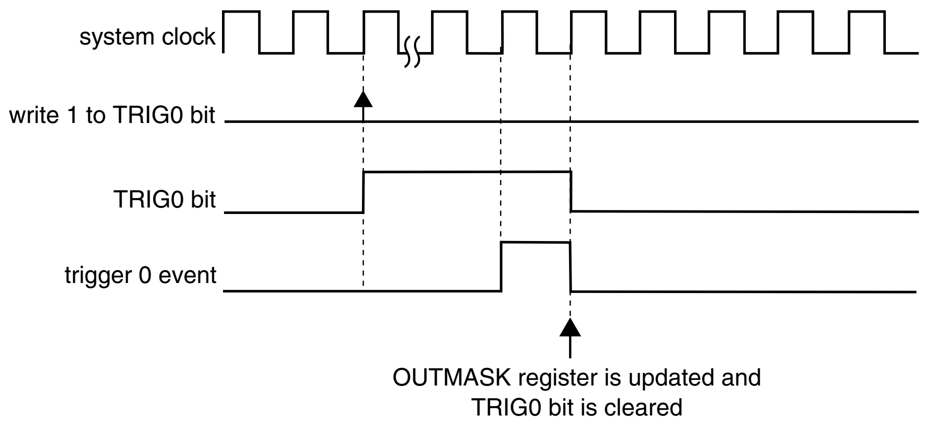


Figure 39-257. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 1), then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger. An example with a hardware trigger follows.

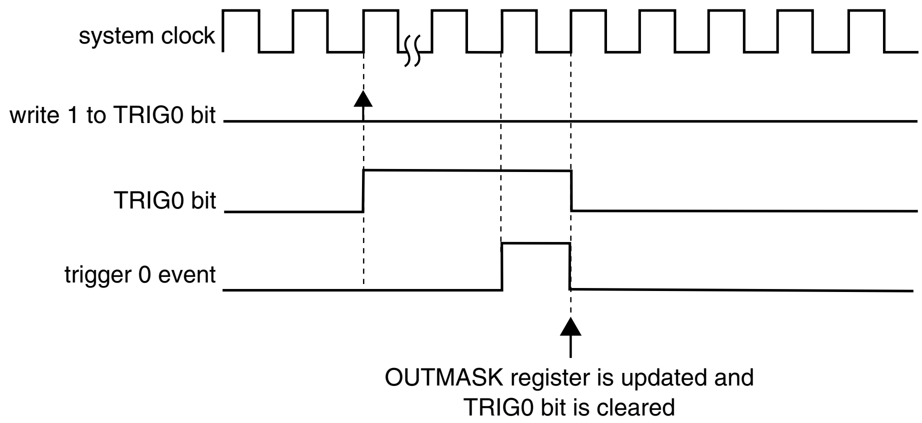


Figure 39-258. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 1), and a hardware trigger was used

39.4.11.8 INVCTRL register synchronization

The INVCTRL register synchronization updates the INVCTRL register with its buffer value.

The INVCTRL register can be updated at each rising edge of system clock (INVC = 0) or by the enhanced PWM synchronization (INVC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the INVCTRL register synchronization depends on SWINVC and HWINVC bits.

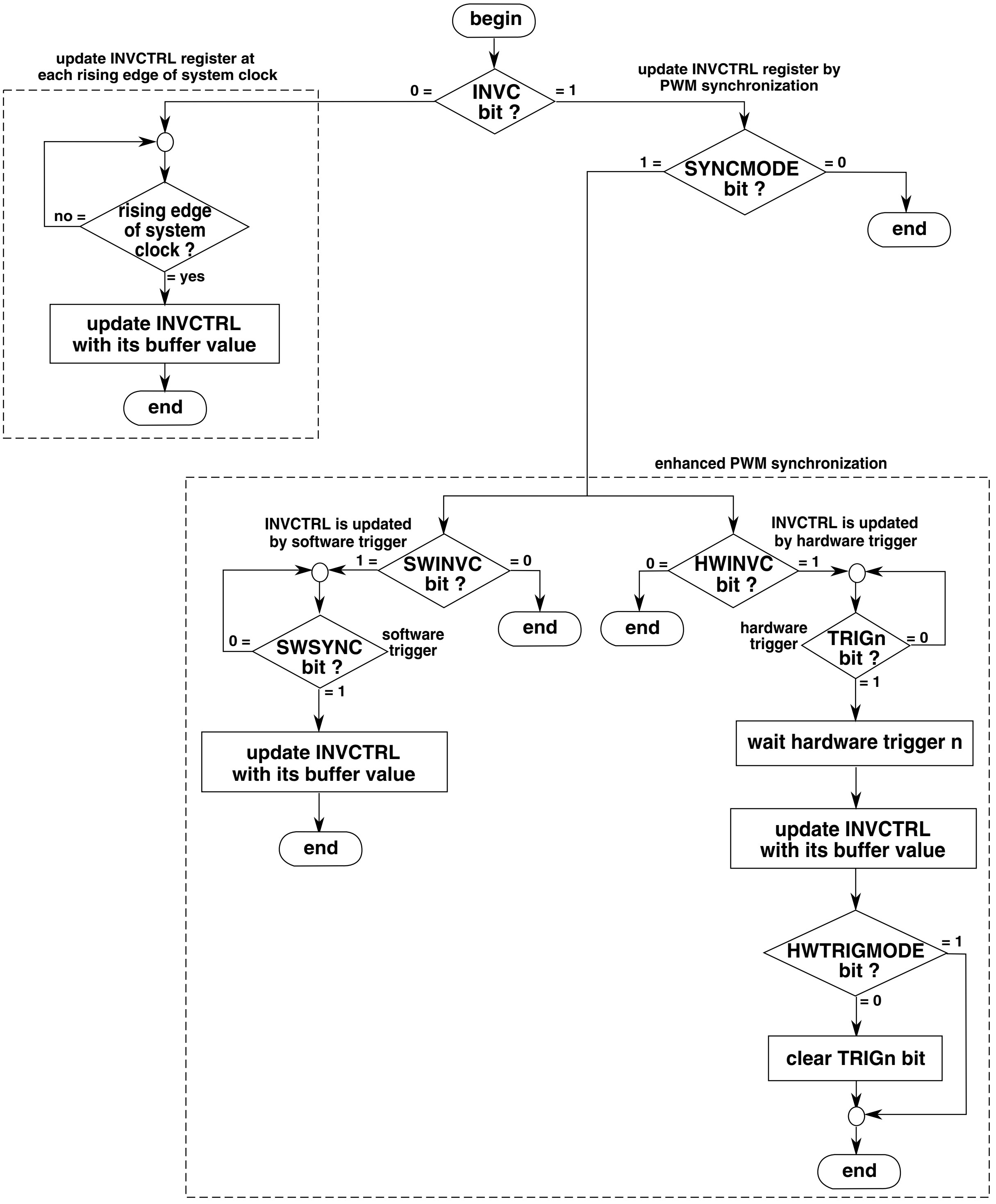


Figure 39-259. INVCTRL register synchronization flowchart

39.4.11.9 SWOCTRL register synchronization

The SWOCTRL register synchronization updates the SWOCTRL register with its buffer value.

The SWOCTRL register can be updated at each rising edge of system clock (SWOC = 0) or by the enhanced PWM synchronization (SWOC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the SWOCTRL register synchronization depends on SWSOC and HWSOC bits.

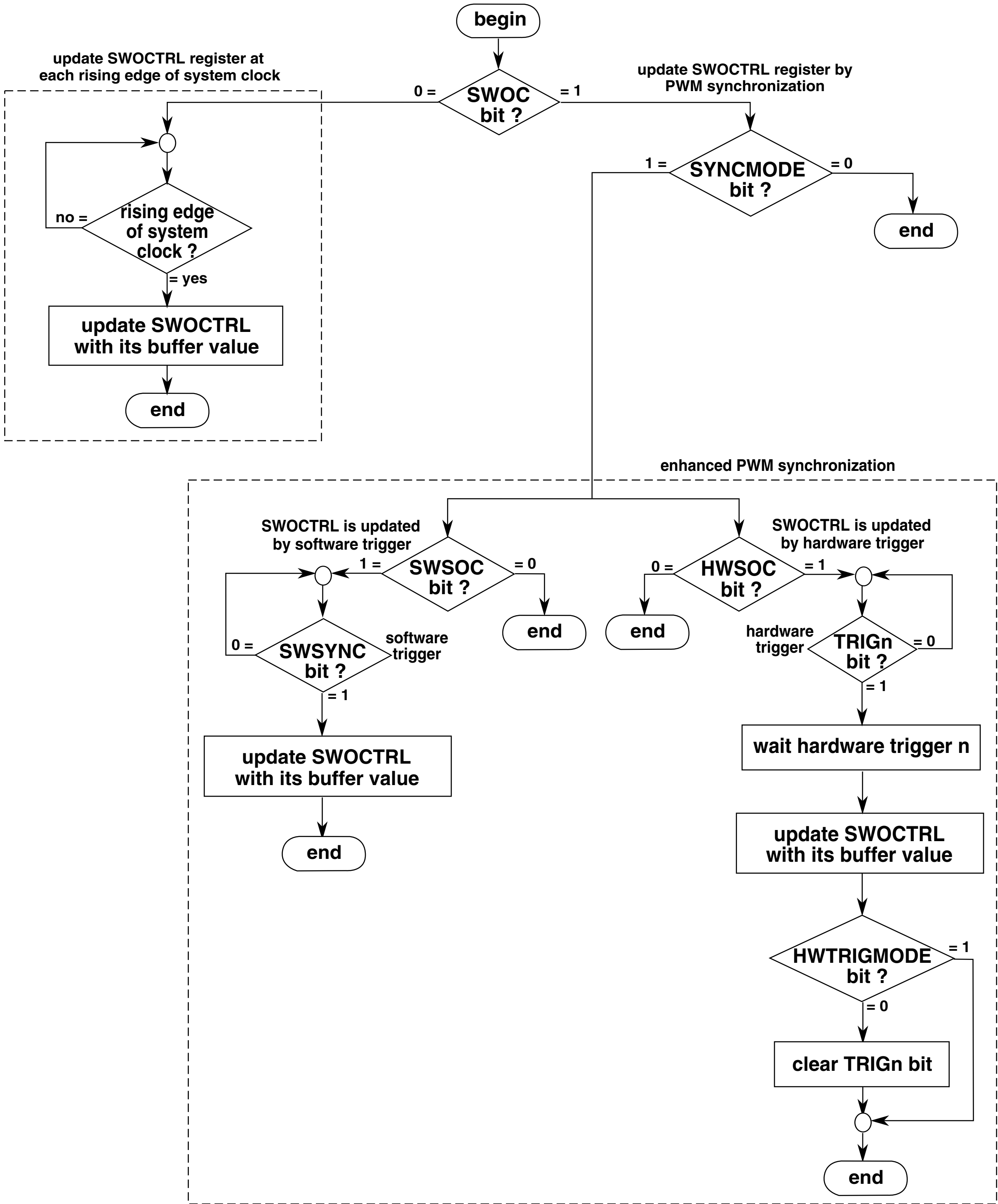


Figure 39-260. SWOCTRL register synchronization flowchart

39.4.11.10 FTM counter synchronization

The FTM counter synchronization is a mechanism that allows the FTM to restart the PWM generation at a certain point in the PWM period. The channels outputs are forced to their initial value, except for channels in Output Compare mode, and the FTM counter is forced to its initial counting value defined by CNTIN register.

The following figure shows the FTM counter synchronization. Note that after the synchronization event occurs, the channel (n) is set to its initial value and the channel (n +1) is not set to its initial value due to a specific timing of this figure in which the deadtime insertion prevents this channel output from transitioning to 1. If no deadtime insertion is selected, then the channel (n+1) transitions to logical value 1 immediately after the synchronization event occurs.

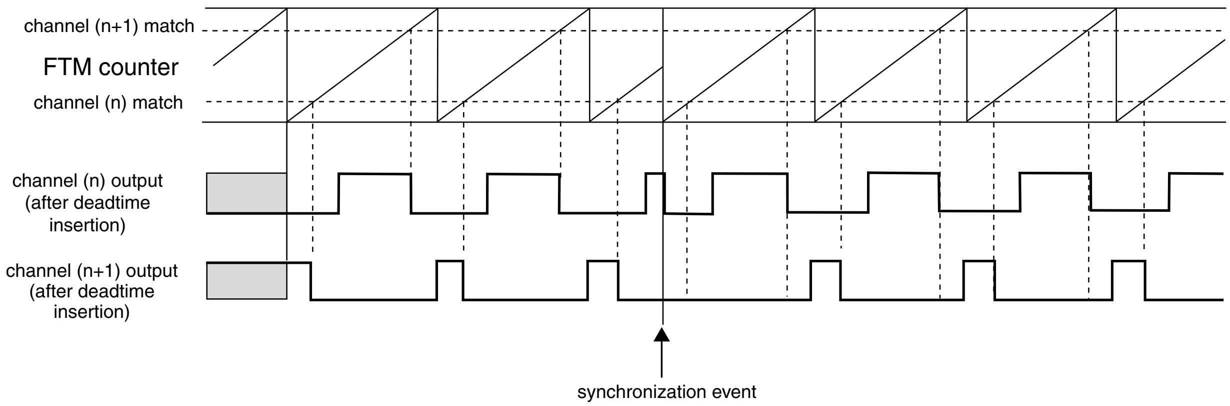


Figure 39-261. FTM counter synchronization

The FTM counter synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, the FTM counter must be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the FTM counter synchronization depends on SWRSTCNT and HWRSTCNT bits according to the following flowchart.

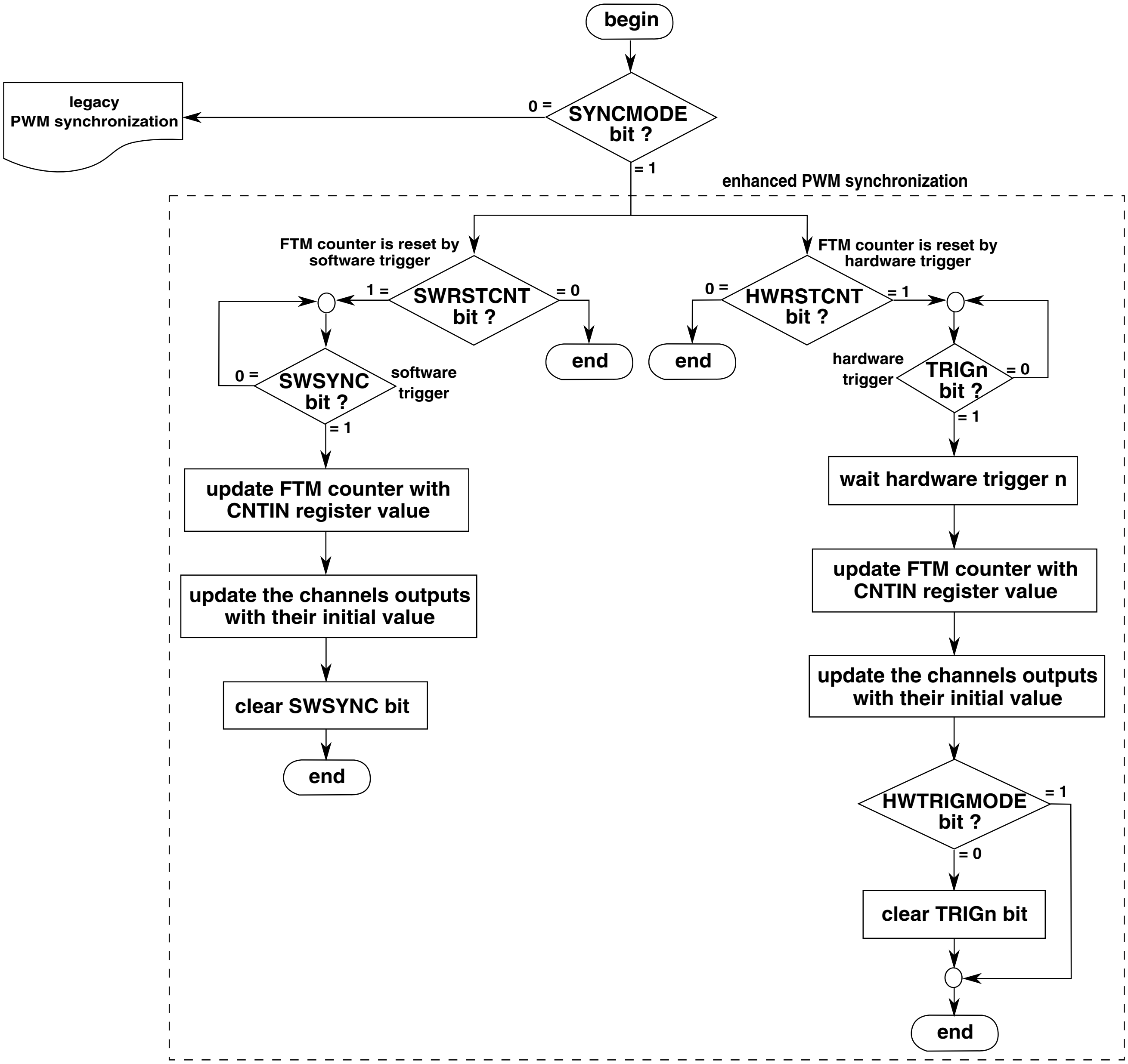


Figure 39-262. FTM counter synchronization flowchart

In the case of legacy PWM synchronization, the FTM counter synchronization depends on REINIT and PWMSYNC bits according to the following description.

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 0) then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

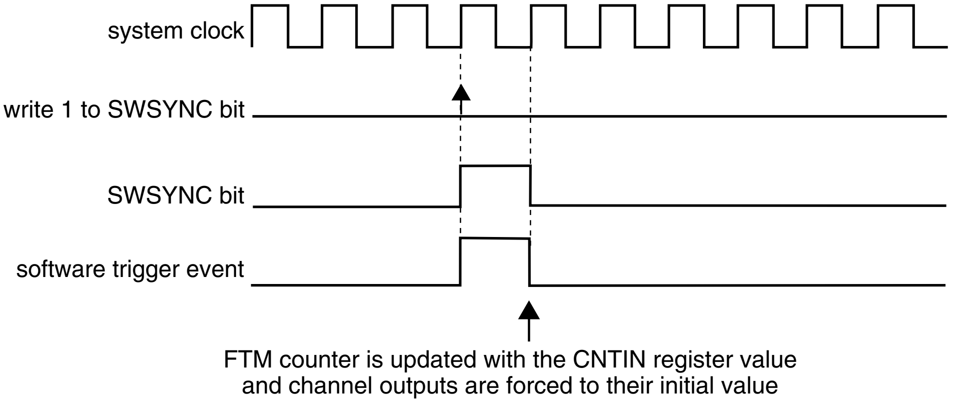


Figure 39-263. FTM counter synchronization with (SYNCMODE = 0), (REINIT = 1), (PWMSYNC = 0), and software trigger was used

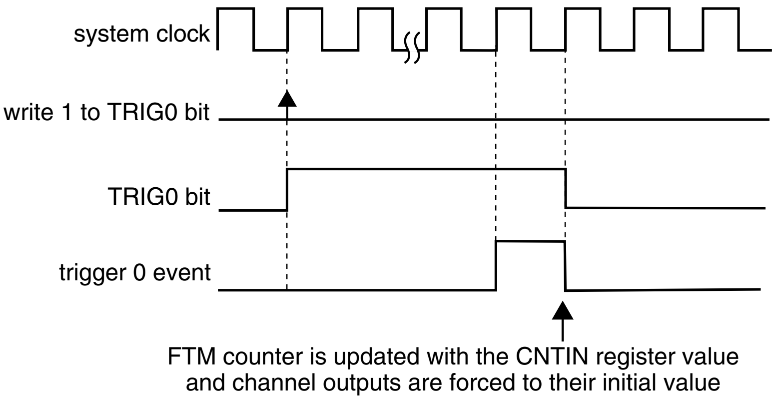


Figure 39-264. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 1) then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger.

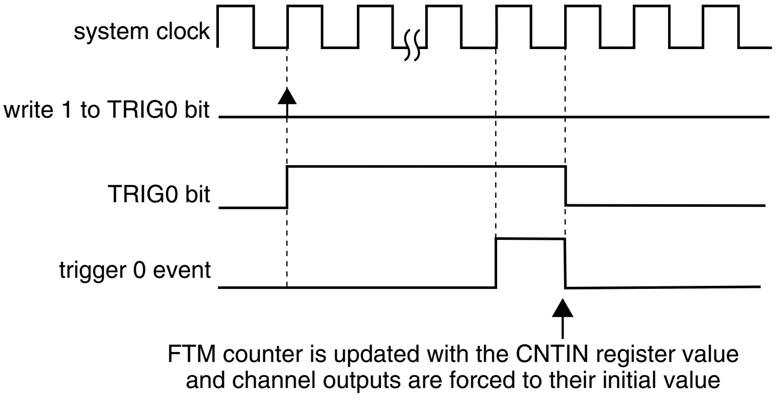


Figure 39-265. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 1), and a hardware trigger was used

39.4.12 Inverting

The invert functionality swaps the signals between channel (n) and channel (n+1) outputs. The inverting operation is selected when:

- FTMEN = 1
- QUADEN = 0
- DECAPEN = 0
- COMBINE = 1
- COMP = 1
- CPWMS = 0, and
- INVm = 1 (where m represents a channel pair)

The INVm bit in INVCTRL register is updated with its buffer value according to INVCTRL register synchronization

In High-True (ELSnB:ELSnA = 1:0) Combine mode, the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN), forced high at the channel (n) match and forced low at the channel (n+1) match. If the inverting is selected, the channel (n) output behavior is changed to force high at the beginning of the PWM period, force low at the channel (n) match and force high at the channel (n+1) match. See the following figure.

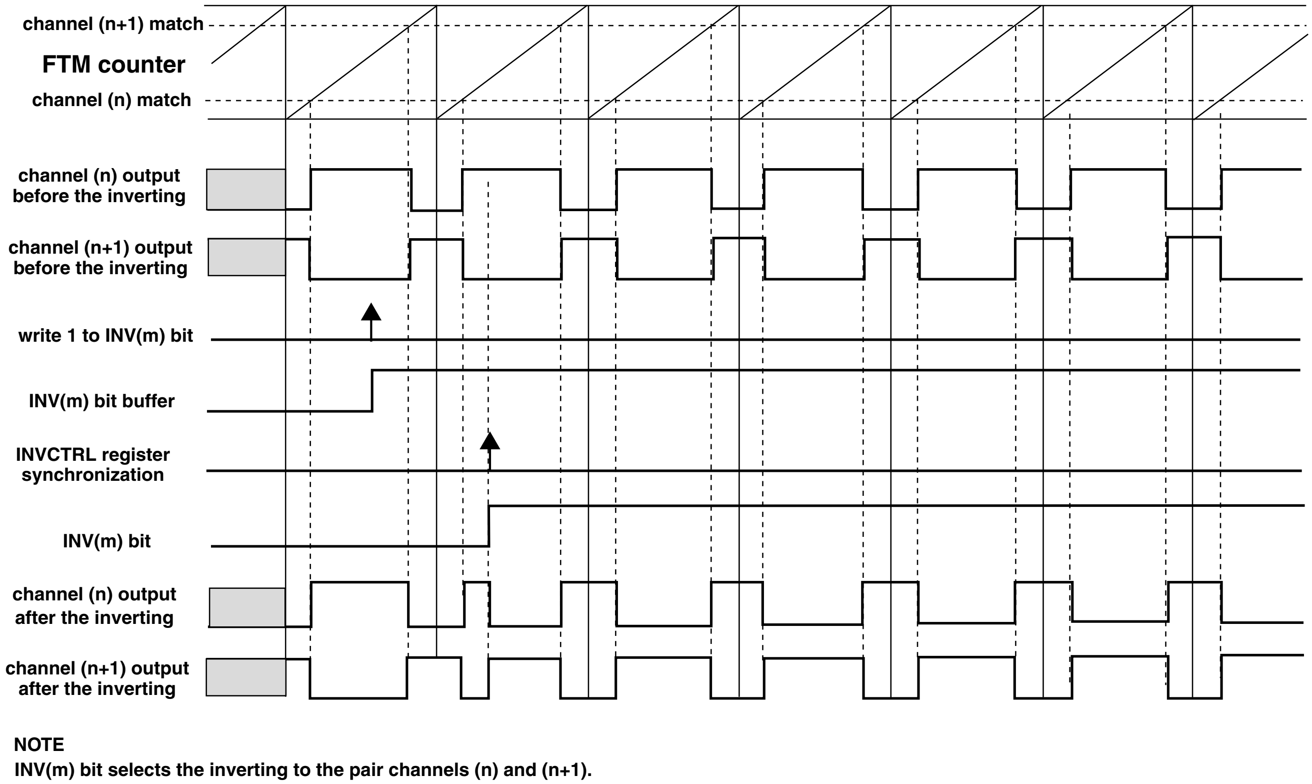
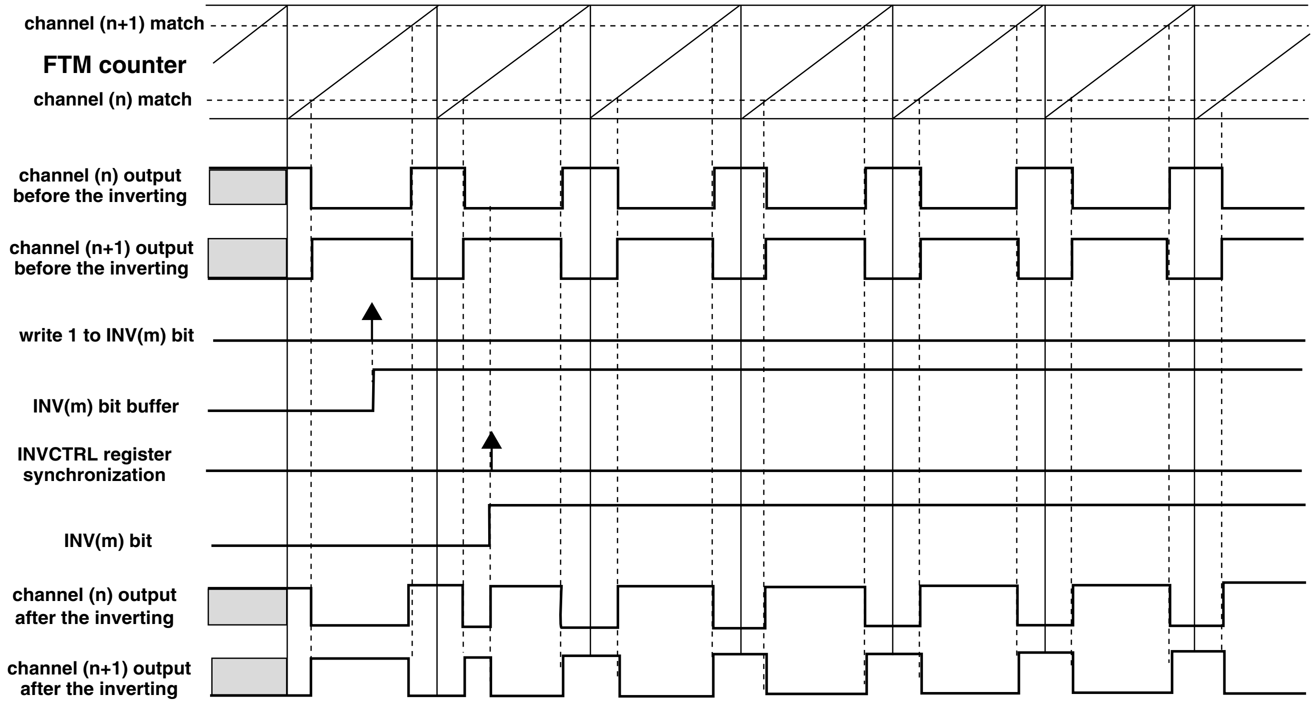


Figure 39-266. Channels (n) and (n+1) outputs after the inverting in High-True (ELSnB:ELSnA = 1:0) Combine mode

Note that the ELSnB:ELSnA bits value should be considered because they define the active state of the channels outputs. In Low-True (ELSnB:ELSnA = X:1) Combine mode, the channel (n) output is forced high at the beginning of the period, forced low at the channel (n) match and forced high at the channel (n+1) match. When inverting is selected, the channels (n) and (n+1) present waveforms as shown in the following figure.



NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Figure 39-267. Channels (n) and (n+1) outputs after the inverting in Low-True (ELSnB:ELSnA = X:1) Combine mode

Note

The inverting feature must be used only in Combine mode.

39.4.13 Software output control

The software output control forces the channel output according to software defined values at a specific time in the PWM generation.

The software output control is selected when:

- FTMEN = 1
- QUADEN = 0
- DECAPEN = 0
- COMBINE = 1
- CPWMS = 0, and
- CHnOC = 1

The CHnOC bit enables the software output control for a specific channel output and the CHnOCV selects the value that is forced to this channel output.

Both CHnOC and CHnOCV bits in SWOCTRL register are buffered and updated with their buffer value according to SWOCTRL register synchronization.

The following figure shows the channels (n) and (n+1) outputs signals when the software output control is used. In this case the channels (n) and (n+1) are set to Combine and Complementary mode.

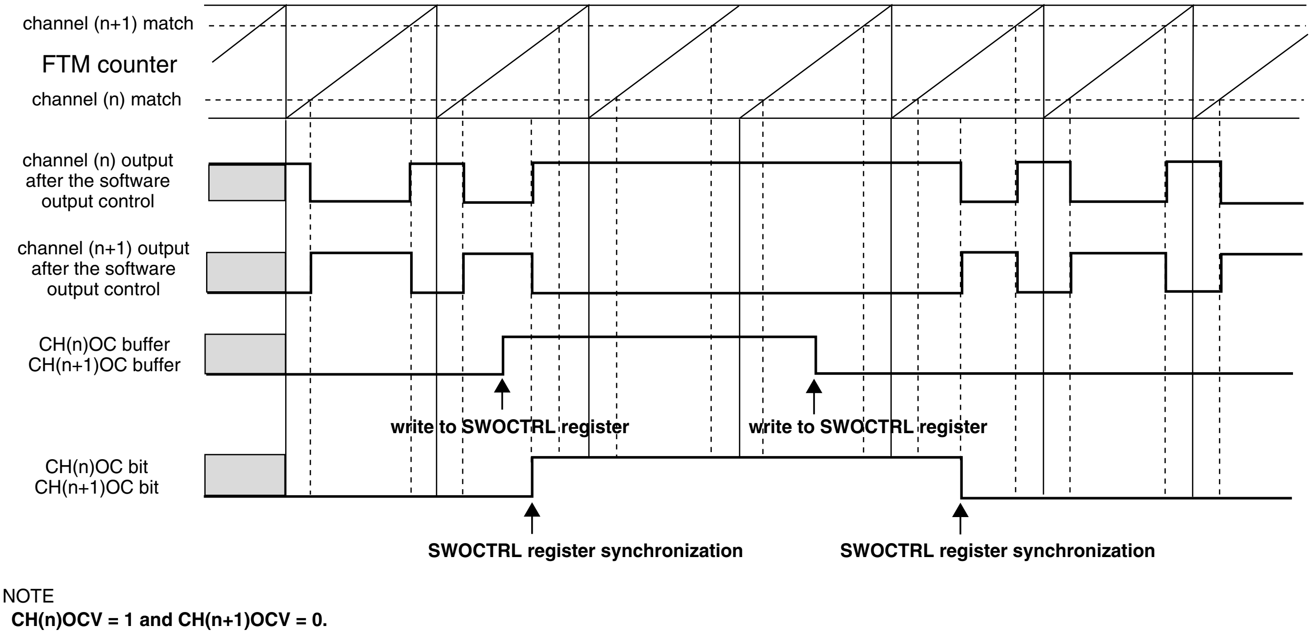


Figure 39-268. Example of software output control in Combine and Complementary mode

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is zero.

Table 39-306. Software ouput control behavior when (COMP = 0)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to one

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is one.

Table 39-307. Software ouput control behavior when (COMP = 1)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to zero

Note

- The software output control feature must be used only in Combine mode.
- The CH(n)OC and CH(n+1)OC bits should be equal.
- The COMP bit must not be modified when software output control is enabled, that is, CH(n)OC = 1 and/or CH(n +1)OC = 1.
- Software output control has the same behavior with disabled or enabled FTM counter (see the CLKS field description in the Status and Control register).

39.4.14 Deadtime insertion

The deadtime insertion is enabled when (DTEN = 1) and (DTVAL[5:0] is non- zero).

DEADTIME register defines the deadtime delay that can be used for all FTM channels. The DTPS[1:0] bits define the prescaler for the system clock and the DTVAL[5:0] bits define the deadtime modulo, that is, the number of the deadtime prescaler clocks.

The deadtime delay insertion ensures that no two complementary signals (channels (n) and (n+1)) drive the active state at the same time.

If POL(n) = 0, POL(n+1) = 0, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the low value until the end of the deadtime delay when the channel (n) output is set. Similarly, when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the low value until the end of the deadtime delay when the channel (n+1) output is set. See the following figures.

If POL(n) = 1, POL(n+1) = 1, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the high value until the end of the deadtime delay when the channel (n) output is cleared. Similarly, when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the high value until the end of the deadtime delay when the channel (n +1) output is cleared.

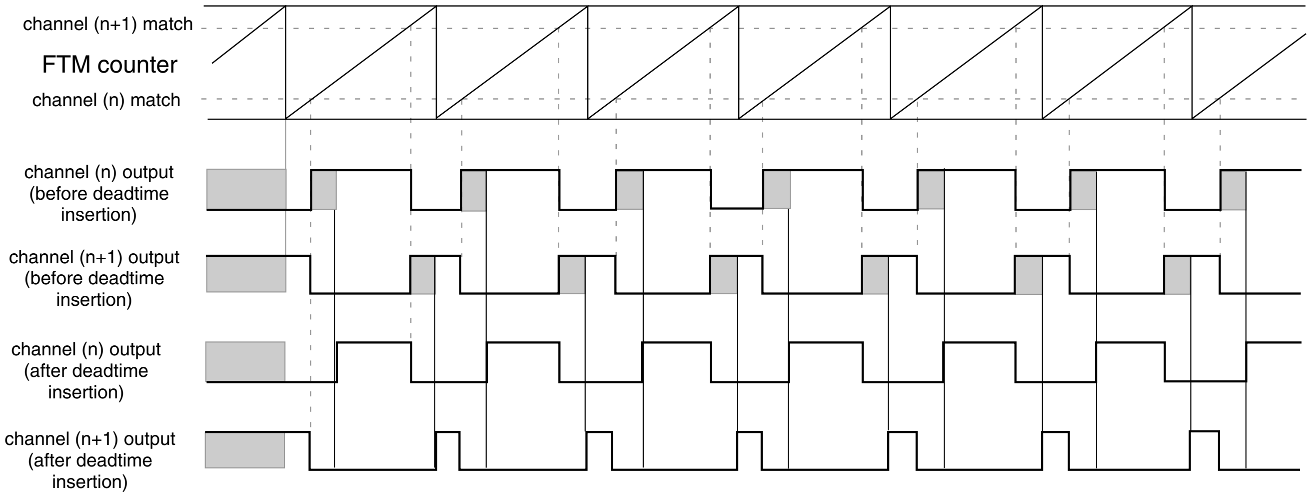


Figure 39-269. Deadtime insertion with ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0

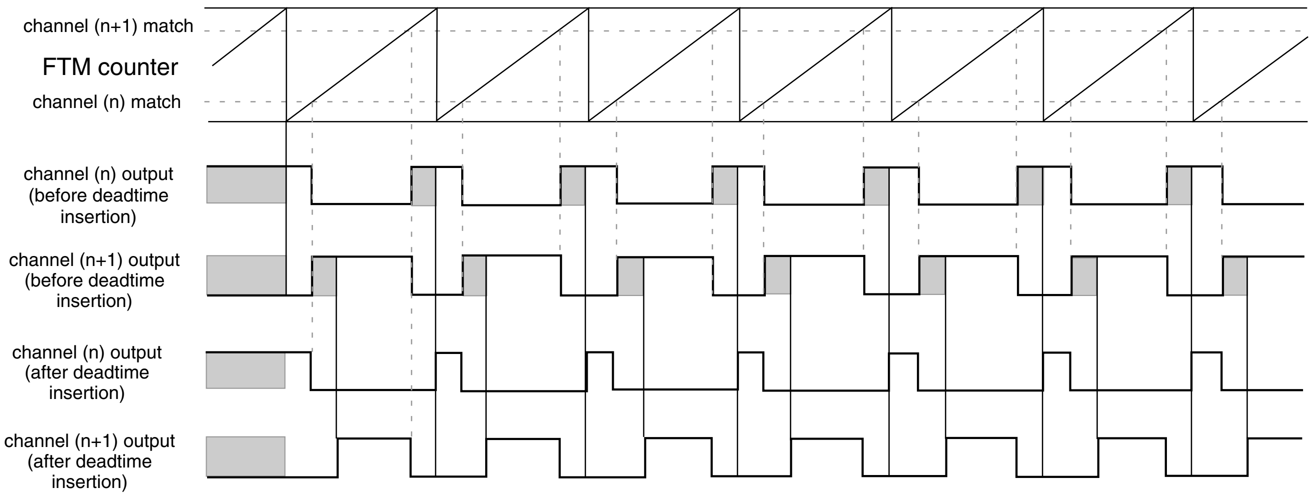


Figure 39-270. Deadtime insertion with ELSnB:ELSnA = X:1, POL(n) = 0, and POL(n+1) = 0

NOTE

The deadtime feature must be used only in Combine and Complementary modes.

39.4.14.1 Deadtime insertion corner cases

If (PS[2:0] is cleared), (DTPS[1:0] = 0:0 or DTPS[1:0] = 0:1):

- and the deadtime delay is greater than or equal to the channel (n) duty cycle ((C(n +1)V – C(n)V) × system clock), then the channel (n) output is always the inactive value (POL(n) bit value).
- and the deadtime delay is greater than or equal to the channel (n+1) duty cycle ((MOD – CNTIN + 1 – (C(n+1)V – C(n)V)) × system clock), then the channel (n+1) output is always the inactive value (POL(n+1) bit value).

Although, in most cases the deadtime delay is not comparable to channels (n) and (n+1) duty cycle, the following figures show examples where the deadtime delay is comparable to the duty cycle.

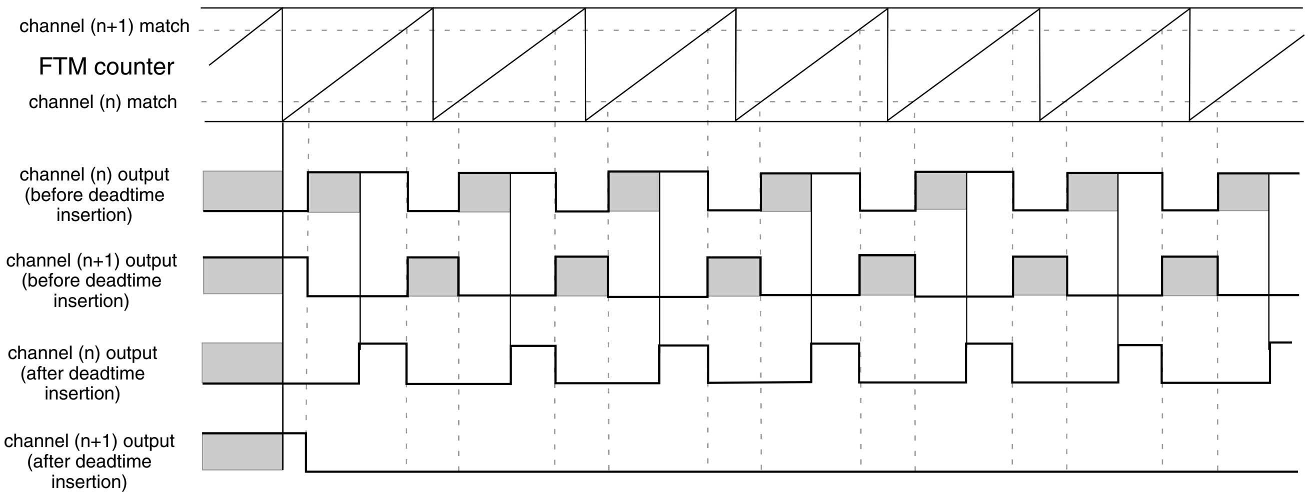


Figure 39-271. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channel (n+1) duty cycle

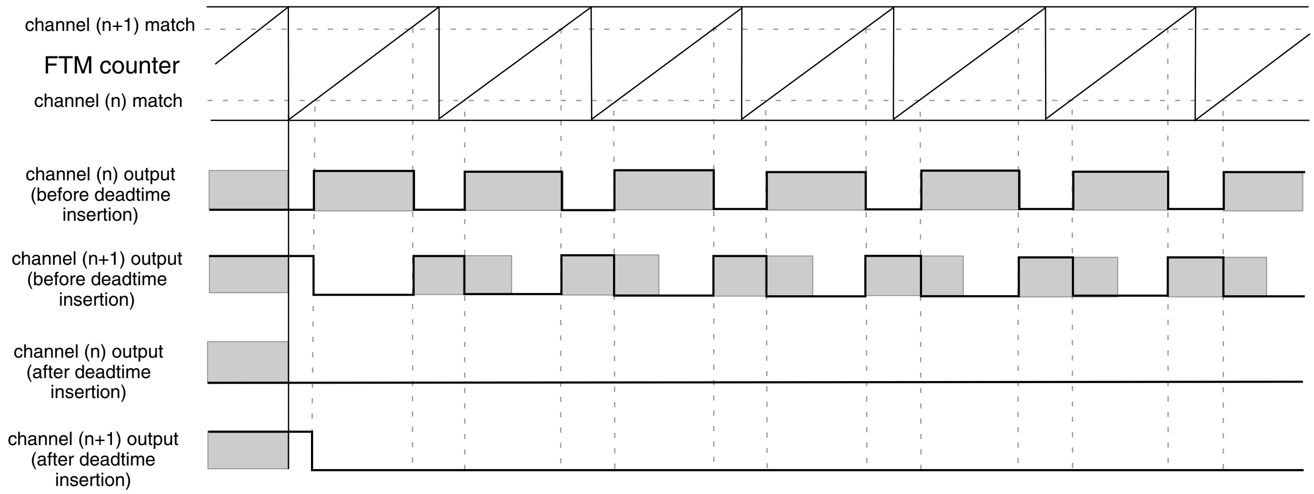


Figure 39-272. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channels (n) and (n+1) duty cycle

39.4.15 Output mask

The output mask can be used to force channels output to their inactive state through software. For example: to control a BLDC motor.

Any write to the OUTMASK register updates its write buffer. The OUTMASK register is updated with its buffer value by PWM synchronization; see OUTMASK register synchronization.

If CHnOM = 1, then the channel (n) output is forced to its inactive state (POLn bit value). If CHnOM = 0, then the channel (n) output is unaffected by the output mask. See the following figure.

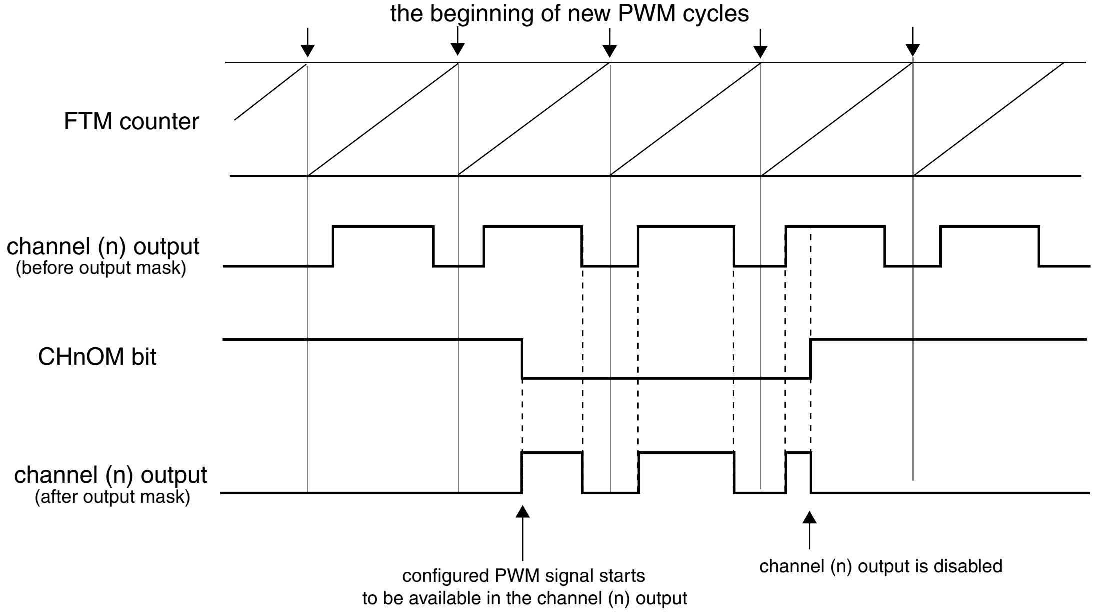


Figure 39-273. Output mask with POLn = 0

The following table shows the output mask result before the polarity control.

Table 39-308. Output mask result for channel (n) before the polarity control

CHnOM	Output Mask Input	Output Mask Result
0	inactive state	inactive state
	active state	active state
1	inactive state	inactive state
	active state	

Note

The output mask feature must be used only in Combine mode.

39.4.16 Fault control

The fault control is enabled if (FTMEN = 1) and (FAULTM[1:0] ≠ 0:0).

FTM can have up to four fault inputs. FAULTnEN bit (where n = 0, 1, 2, 3) enables the fault input n and FFLTRnEN bit enables the fault input n filter. FFVAL[3:0] bits select the value of the enabled filter in each enabled fault input.

First, each fault input signal is synchronized by the system clock; see the synchronizer block in the following figure. Following synchronization, the fault input n signal enters the filter block. When there is a state change in the fault input n signal, the 5-bit counter

is reset and starts counting up. As long as the new state is stable on the fault input n, the counter continues to increment. If the 5-bit counter overflows, that is, the counter exceeds the value of the FFVAL[3:0] bits, the new fault input n value is validated. It is then transmitted as a pulse edge to the edge detector.

If the opposite edge appears on the fault input n signal before validation (counter overflow), the counter is reset. At the next input transition, the counter starts counting again. Any pulse that is shorter than the minimum value selected by FFVAL[3:0] bits (× system clock) is regarded as a glitch and is not passed on to the edge detector.

The fault input n filter is disabled when the FFVAL[3:0] bits are zero or when FAULTnEN = 0. In this case, the fault input n signal is delayed 2 rising edges of the system clock and the FAULTFn bit is set on 3th rising edge of the system clock after a rising edge occurs on the fault input n.

If FFVAL[3:0] ≠ 0000 and FAULTnEN = 1, then the fault input n signal is delayed (3 + FFVAL[3:0]) rising edges of the system clock, that is, the FAULTFn bit is set (4 + FFVAL[3:0]) rising edges of the system clock after a rising edge occurs on the fault input n.

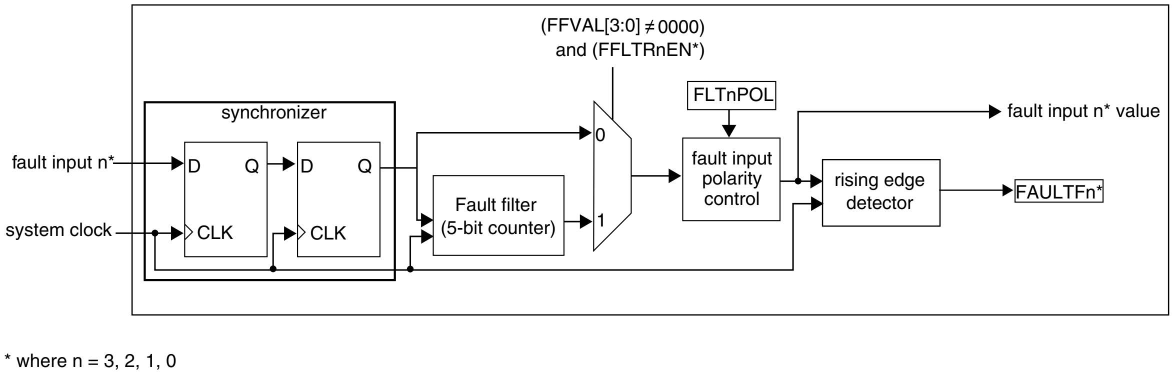


Figure 39-274. Fault input n control block diagram

If the fault control and fault input n are enabled and a rising edge at the fault input n signal is detected, a fault condition has occurred and the FAULTFn bit is set. The FAULTF bit is the logic OR of FAULTFn[3:0] bits. See the following figure.

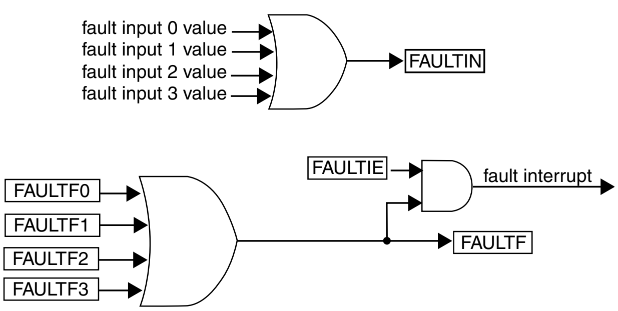


Figure 39-275. FAULTF and FAULTIN bits and fault interrupt

If the fault control is enabled (FAULTM[1:0] ≠ 0:0), a fault condition has occurred and (FAULTEN = 1), then outputs are forced to their safe values:

- Channel (n) output takes the value of POL(n)
- Channel (n+1) takes the value of POL(n+1)

The fault interrupt is generated when (FAULTF = 1) and (FAULTIE = 1). This interrupt request remains set until:

- Software clears the FAULTF bit by reading FAULTF bit as 1 and writing 0 to it
- Software clears the FAULTIE bit
- A reset occurs

Note

The fault control must be used only in Combine mode.

39.4.16.1 Automatic fault clearing

If the automatic fault clearing is selected (FAULTM[1:0] = 1:1), then the channels output disabled by fault control is again enabled when the fault input signal (FAULTIN) returns to zero and a new PWM cycle begins. See the following figure.

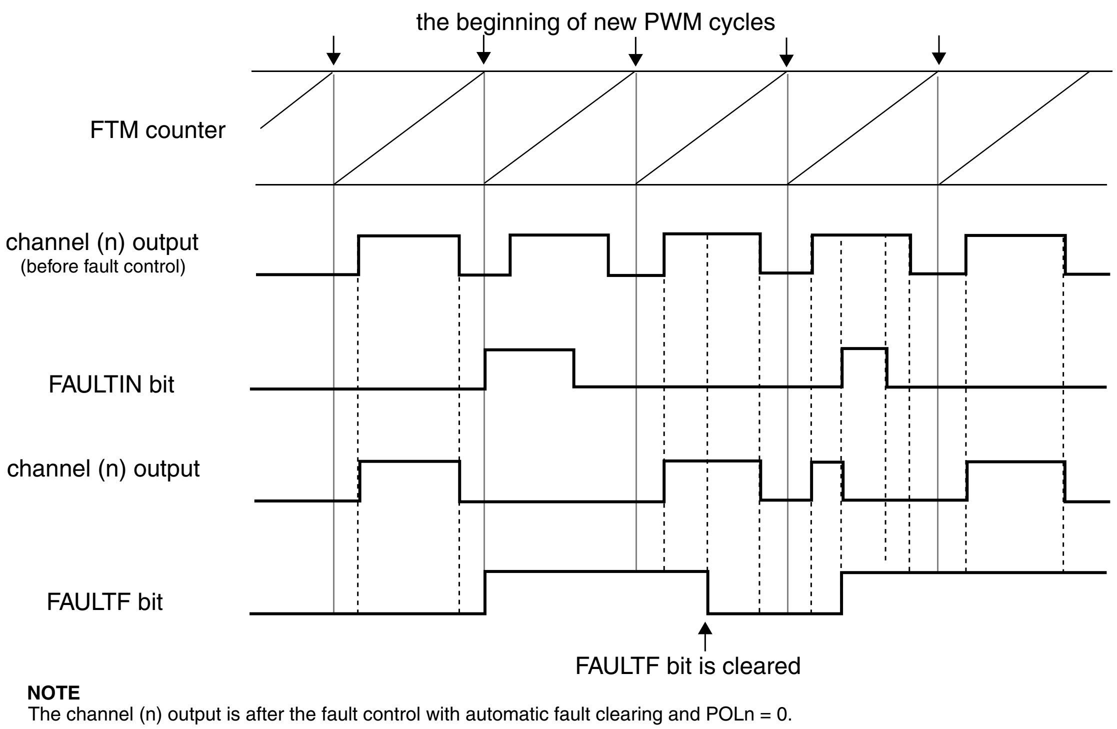


Figure 39-276. Fault control with automatic fault clearing

39.4.16.2 Manual fault clearing

If the manual fault clearing is selected (FAULTM[1:0] = 0:1 or 1:0), then the channels output disabled by fault control is again enabled when the FAULTF bit is cleared and a new PWM cycle begins. See the following figure.

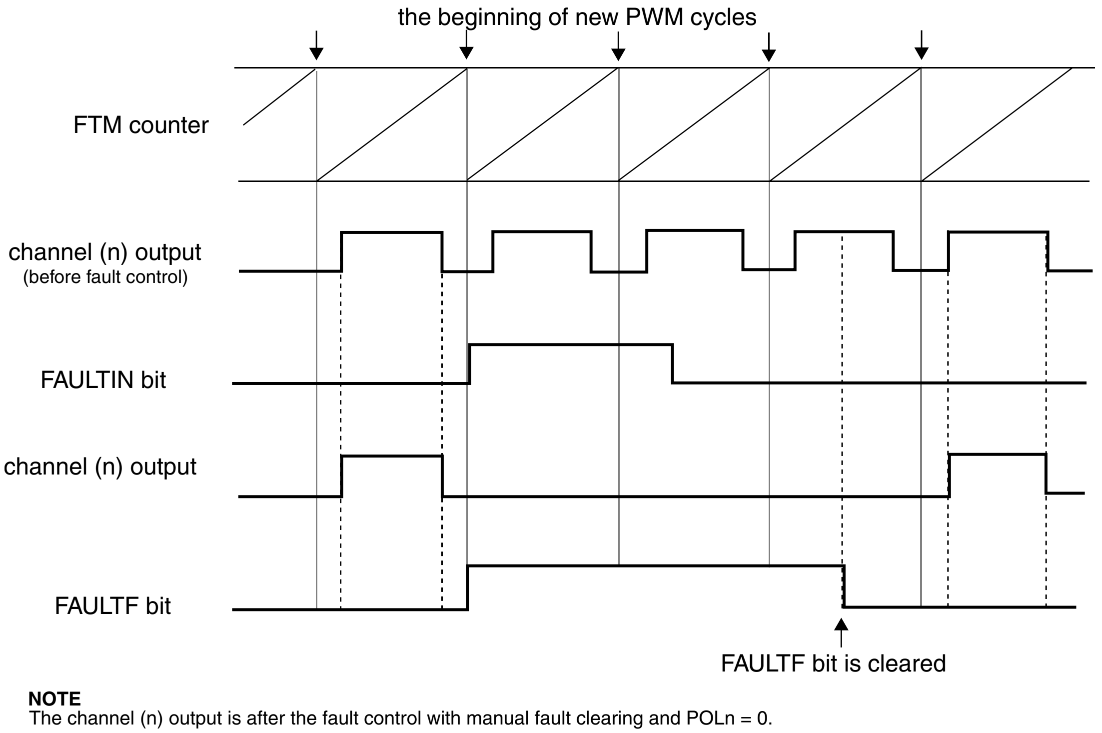


Figure 39-277. Fault control with manual fault clearing

39.4.16.3 Fault inputs polarity control

The FLTjPOL bit selects the fault input j polarity, where j = 0, 1, 2, 3:

- If FLTjPOL = 0, the fault j input polarity is high, so the logical one at the fault input j indicates a fault.
- If FLTjPOL = 1, the fault j input polarity is low, so the logical zero at the fault input j indicates a fault.

39.4.17 Polarity control

The POLn bit selects the channel (n) output polarity:

- If POLn = 0, the channel (n) output polarity is high, so the logical one is the active state and the logical zero is the inactive state.
- If POLn = 1, the channel (n) output polarity is low, so the logical zero is the active state and the logical one is the inactive state.

Note

The polarity control must be used only in Combine mode.

39.4.18 Initialization

The initialization forces the CHnOI bit value to the channel (n) output when a one is written to the INIT bit.

The initialization depends on COMP and DTEN bits. The following table shows the values that channels (n) and (n+1) are forced by initialization when the COMP and DTEN bits are zero.

Table 39-309. Initialization behavior when (COMP = 0 and DTEN = 0)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	0	is forced to zero	is forced to zero
0	1	is forced to zero	is forced to one
1	0	is forced to one	is forced to zero
1	1	is forced to one	is forced to one

The following table shows the values that channels (n) and (n+1) are forced by initialization when (COMP = 1) or (DTEN = 1).

Table 39-310. Initialization behavior when (COMP = 1 or DTEN = 1)

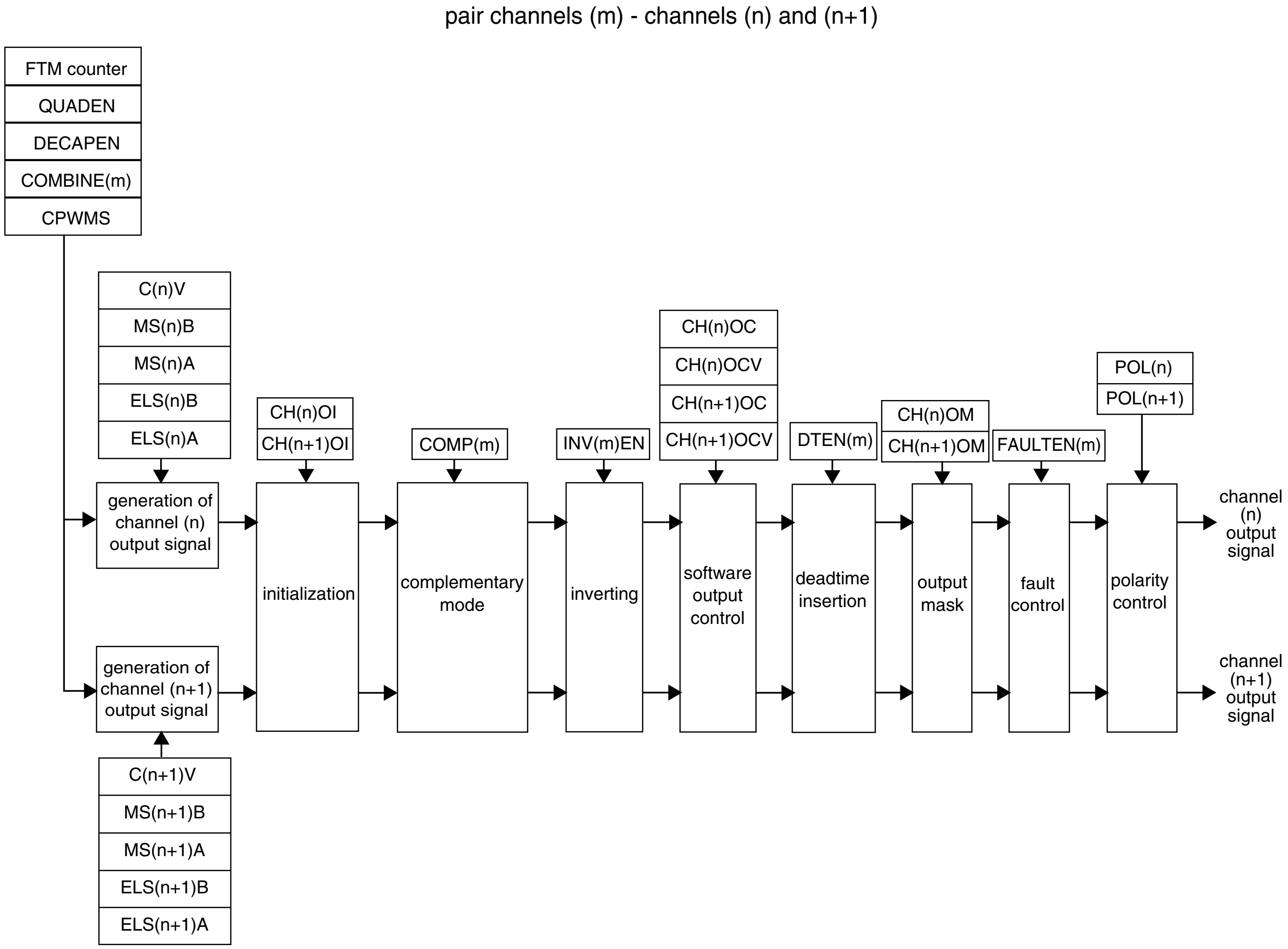
CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	X	is forced to zero	is forced to one
1	X	is forced to one	is forced to zero

Note

The initialization feature must be used only in Combine mode and with disabled FTM counter. See the description of the CLKS field in the Status and Control register.

39.4.19 Features priority

The following figure shows the priority of the features used at the generation of channels (n) and (n+1) outputs signals.



NOTE
The channels (n) and (n+1) are in output compare, EPWM, CPWM or combine modes.

Figure 39-278. Priority of the features used at the generation of channels (n) and (n+1) outputs signals

Note

The Initialization feature must not be used with Inverting and Software output control features.

39.4.20 Channel trigger output

If CHjTRIG = 1, where j = 0, 1, 2, 3, 4, or 5, then the FTM generates a trigger when the channel (j) match occurs (FTM counter = C(j)V).

The channel trigger output provides a trigger signal that is used for on-chip modules.

The FTM is able to generate multiple triggers in one PWM period. Because each trigger is generated for a specific channel, several channels are required to implement this functionality. This behavior is described in the following figure.

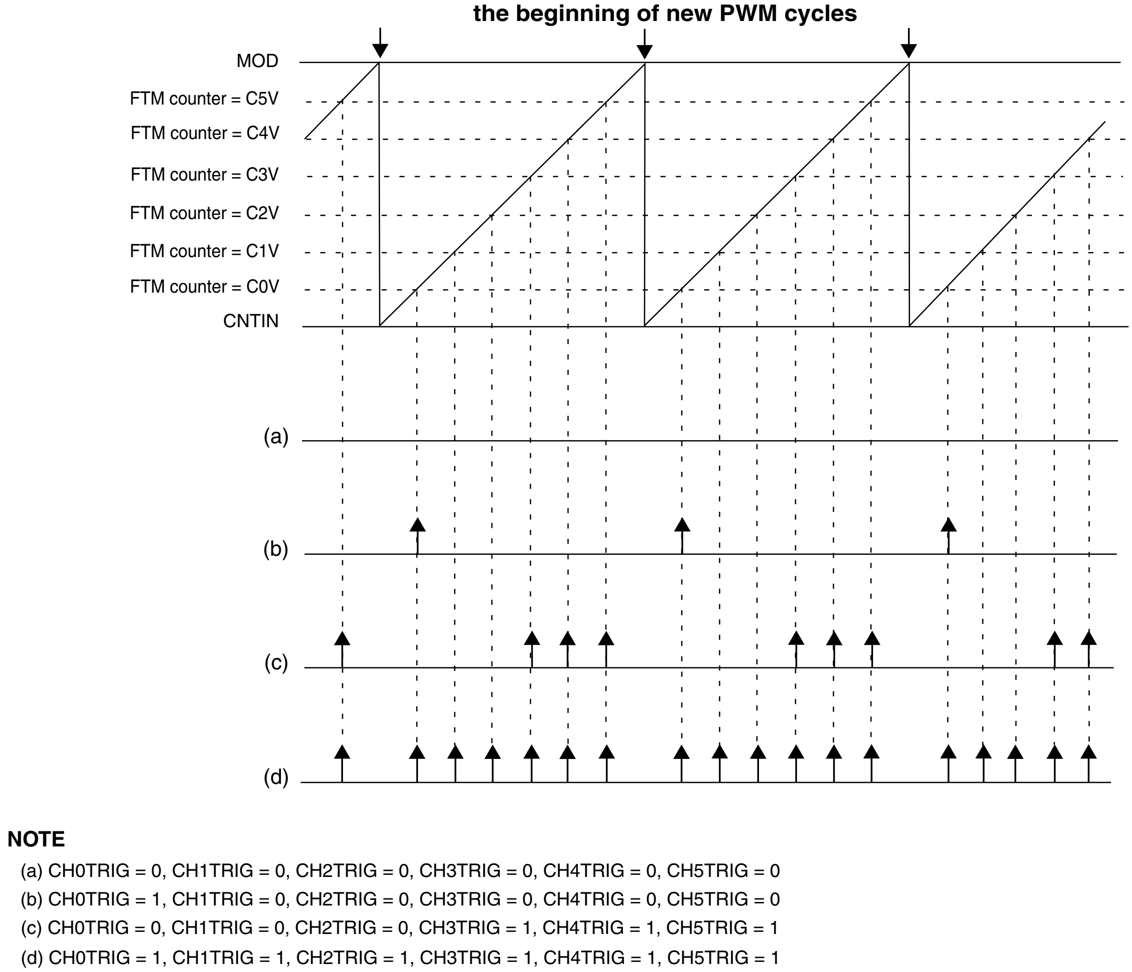


Figure 39-279. Channel match trigger

Note

The channel match trigger must be used only in Combine mode.

39.4.21 Initialization trigger

If INITTRIGEN = 1, then the FTM generates a trigger when the FTM counter is updated with the CNTIN register value in the following cases.

- The FTM counter is automatically updated with the CNTIN register value by the selected counting mode.

- When there is a write to CNT register.
- When there is the FTM counter synchronization.
- If (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits.

The following figures show these cases.

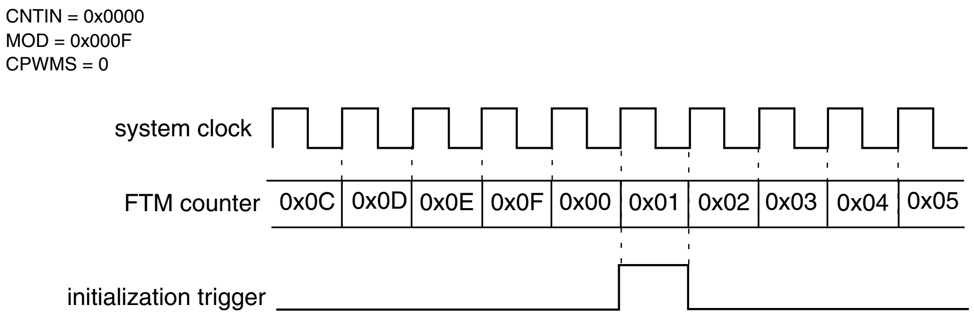


Figure 39-280. Initialization trigger is generated when the FTM counting achieves the CNTIN register value

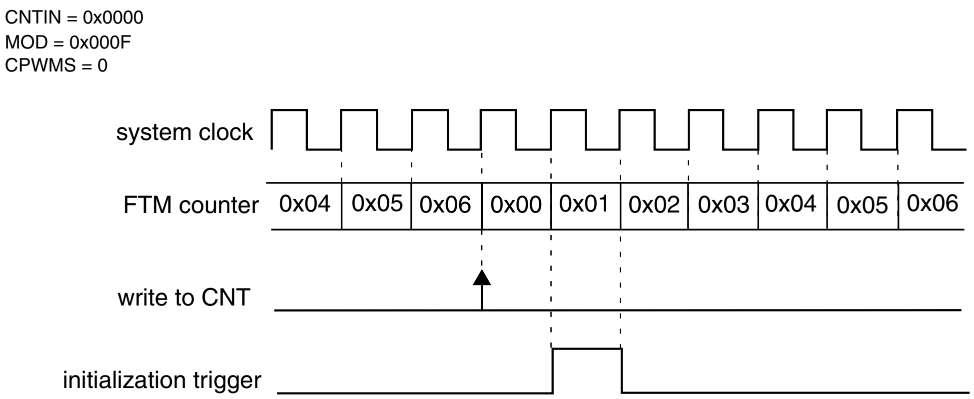


Figure 39-281. Initialization trigger is generated when there is a write to CNT register

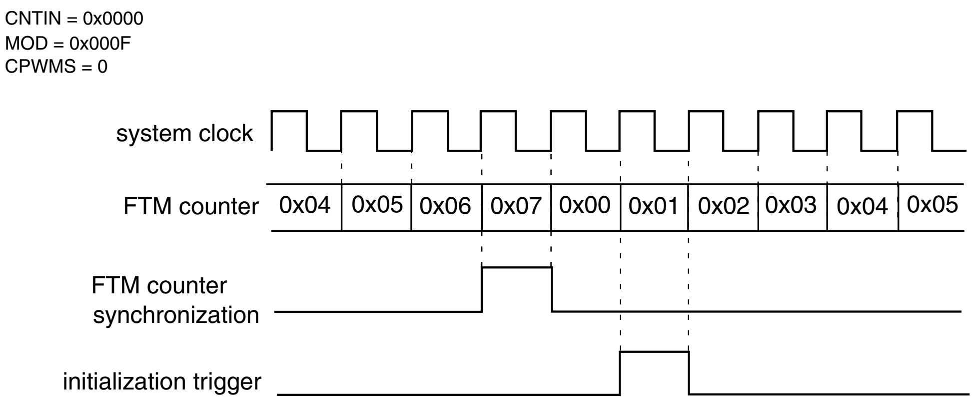


Figure 39-282. Initialization trigger is generated when there is the FTM counter synchronization

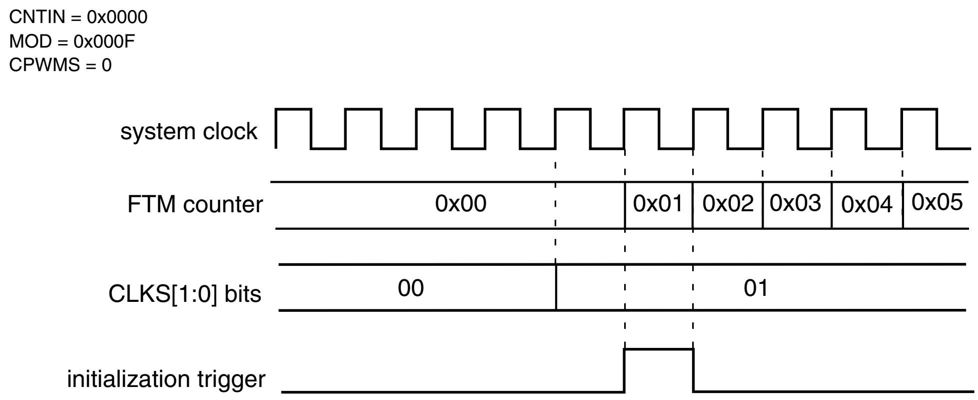


Figure 39-283. Initialization trigger is generated if (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits

The initialization trigger output provides a trigger signal that is used for on-chip modules.

Note

The initialization trigger must be used only in Combine mode.

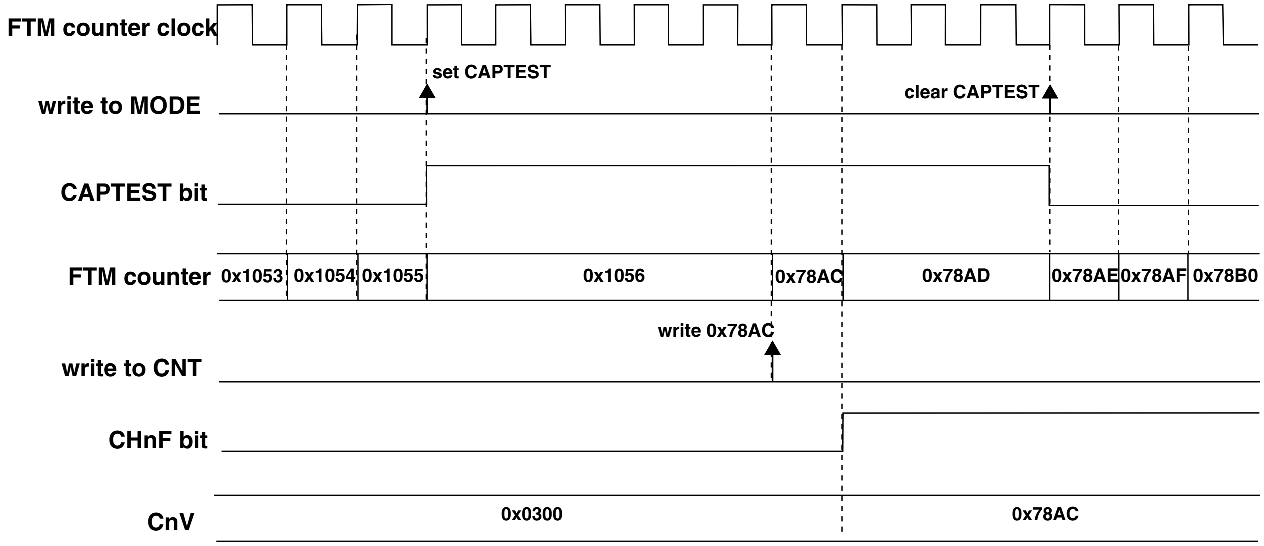
39.4.22 Capture Test mode

The Capture Test mode allows to test the CnV registers, the FTM counter and the interconnection logic between the FTM counter and CnV registers.

In this test mode, all channels must be configured for Input Capture mode and FTM counter must be configured to the Up counting.

When the Capture Test mode is enabled (CAPTEST = 1), the FTM counter is frozen and any write to CNT register updates directly the FTM counter; see the following figure. After it was written, all CnV registers are updated with the written value to CNT register and CHnF bits are set. Therefore, the FTM counter is updated with its next value according to its configuration. Its next value depends on CNTIN, MOD, and the written value to FTM counter.

The next reads of CnV registers return the written value to the FTM counter and the next reads of CNT register return FTM counter next value.



NOTE
- FTM counter configuration: (FTMEN = 1), (QUADEN = 0), (CAPTEST = 1), (CPWMS = 0), (CNTIN = 0x0000), and (MOD = 0xFFFF)
- FTM channel n configuration: input capture mode - (DECAPEN = 0), (COMBINE = 0), and (MSnB:MSnA = 0:0)

Figure 39-284. Capture Test mode

39.4.23 DMA

The channel generates a DMA transfer request according to DMA and CHnIE bits. See the following table.

Table 39-311. Channel DMA transfer request

DMA	CHnIE	Channel DMA Transfer Request	Channel Interrupt
0	0	The channel DMA transfer request is not generated.	The channel interrupt is not generated.
0	1	The channel DMA transfer request is not generated.	The channel interrupt is generated if (CHnF = 1).
1	0	The channel DMA transfer request is not generated.	The channel interrupt is not generated.
1	1	The channel DMA transfer request is generated if (CHnF = 1).	The channel interrupt is not generated.

If DMA = 1, the CHnF bit is cleared either by channel DMA transfer done or reading CnSC while CHnF is set and then writing a zero to CHnF bit according to CHnIE bit. See the following table.

Table 39-312. Clear CHnF bit when DMA = 1

CHnIE	How CHnF Bit Can Be Cleared
0	CHnF bit is cleared either when the channel DMA transfer is done or by reading CnSC while CHnF is set and then writing a 0 to CHnF bit.
1	CHnF bit is cleared when the channel DMA transfer is done.

39.4.24 Dual Edge Capture mode

The Dual Edge Capture mode is selected if FTMEN = 1 and DECAPEN = 1. This mode allows to measure a pulse width or period of the signal on the input of channel (n) of a channel pair. The channel (n) filter can be active in this mode when n is 0 or 2.

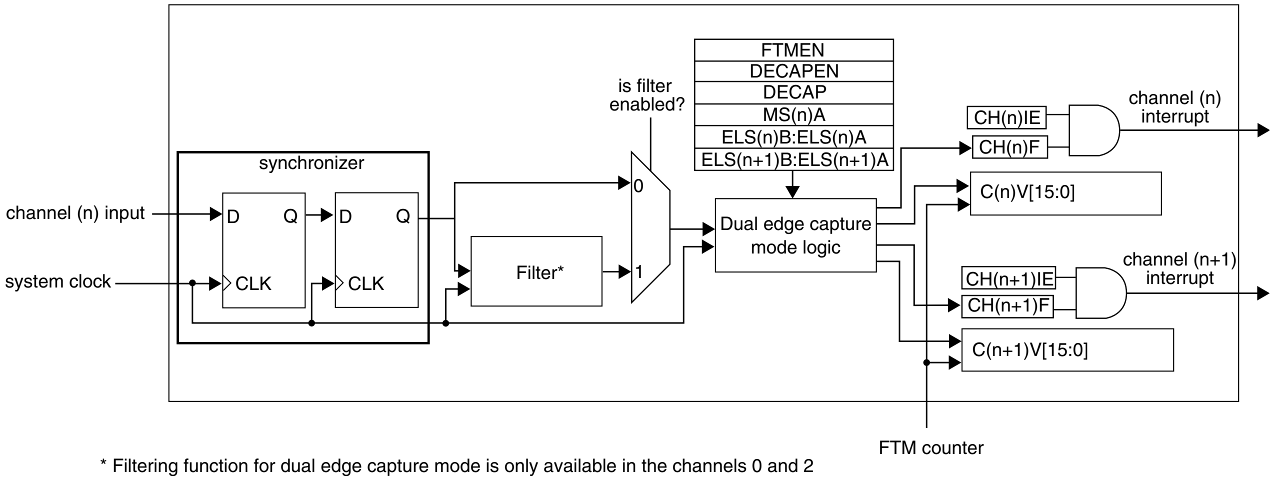


Figure 39-285. Dual Edge Capture mode block diagram

The MS(n)A bit defines if the Dual Edge Capture mode is one-shot or continuous.

The ELS(n)B:ELS(n)A bits select the edge that is captured by channel (n), and ELS(n +1)B:ELS(n+1)A bits select the edge that is captured by channel (n+1). If both ELS(n)B:ELS(n)A and ELS(n+1)B:ELS(n+1)A bits select the same edge, then it is the period measurement. If these bits select different edges, then it is a pulse width measurement.

In the Dual Edge Capture mode, only channel (n) input is used and channel (n+1) input is ignored.

If the selected edge by channel (n) bits is detected at channel (n) input, then CH(n)F bit is set and the channel (n) interrupt is generated (if CH(n)IE = 1). If the selected edge by channel (n+1) bits is detected at channel (n) input and (CH(n)F = 1), then CH(n+1)F bit is set and the channel (n+1) interrupt is generated (if CH(n+1)IE = 1).

The C(n)V register stores the value of FTM counter when the selected edge by channel (n) is detected at channel (n) input. The C(n+1)V register stores the value of FTM counter when the selected edge by channel (n+1) is detected at channel (n) input.

In this mode, a coherency mechanism ensures coherent data when the C(n)V and C(n +1)V registers are read. The only requirement is that C(n)V must be read before C(n +1)V.

Note

- The CH(n)F, CH(n)IE, MS(n)A, ELS(n)B, and ELS(n)A bits are channel (n) bits.
- The CH(n+1)F, CH(n+1)IE, MS(n+1)A, ELS(n+1)B, and ELS(n+1)A bits are channel (n+1) bits.
- The Dual Edge Capture mode must be used with ELS(n)B:ELS(n)A = 0:1 or 1:0, ELS(n+1)B:ELS(n+1)A = 0:1 or 1:0 and the FTM counter in Free running counter.

39.4.24.1 One-Shot Capture mode

The One-Shot Capture mode is selected when (FTMEN = 1), (DECAPEN = 1), and (MS(n)A = 0). In this capture mode, only one pair of edges at the channel (n) input is captured. The ELS(n)B:ELS(n)A bits select the first edge to be captured, and ELS(n +1)B:ELS(n+1)A bits select the second edge to be captured.

The edge captures are enabled while DECAP bit is set. For each new measurement in One-Shot Capture mode, first the CH(n)F and CH(n+1) bits must be cleared, and then the DECAP bit must be set.

In this mode, the DECAP bit is automatically cleared by FTM when the edge selected by channel (n+1) is captured. Therefore, while DECAP bit is set, the one-shot capture is in process. When this bit is cleared, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

Similarly, when the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

39.4.24.2 Continuous Capture mode

The Continuous Capture mode is selected when (FTMEN = 1), (DECAPEN = 1), and (MS(n)A = 1). In this capture mode, the edges at the channel (n) input are captured continuously. The ELS(n)B:ELS(n)A bits select the initial edge to be captured, and ELS(n+1)B:ELS(n+1)A bits select the final edge to be captured.

The edge captures are enabled while DECAP bit is set. For the initial use, first the CH(n)F and CH(n+1)F bits must be cleared, and then DECAP bit must be set to start the continuous measurements.

When the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers. The latest captured values are always available in these registers even after the DECAP bit is cleared.

In this mode, it is possible to clear only the CH(n+1)F bit. Therefore, when the CH(n+1)F bit is set again, the latest captured values are available in C(n)V and C(n+1)V registers.

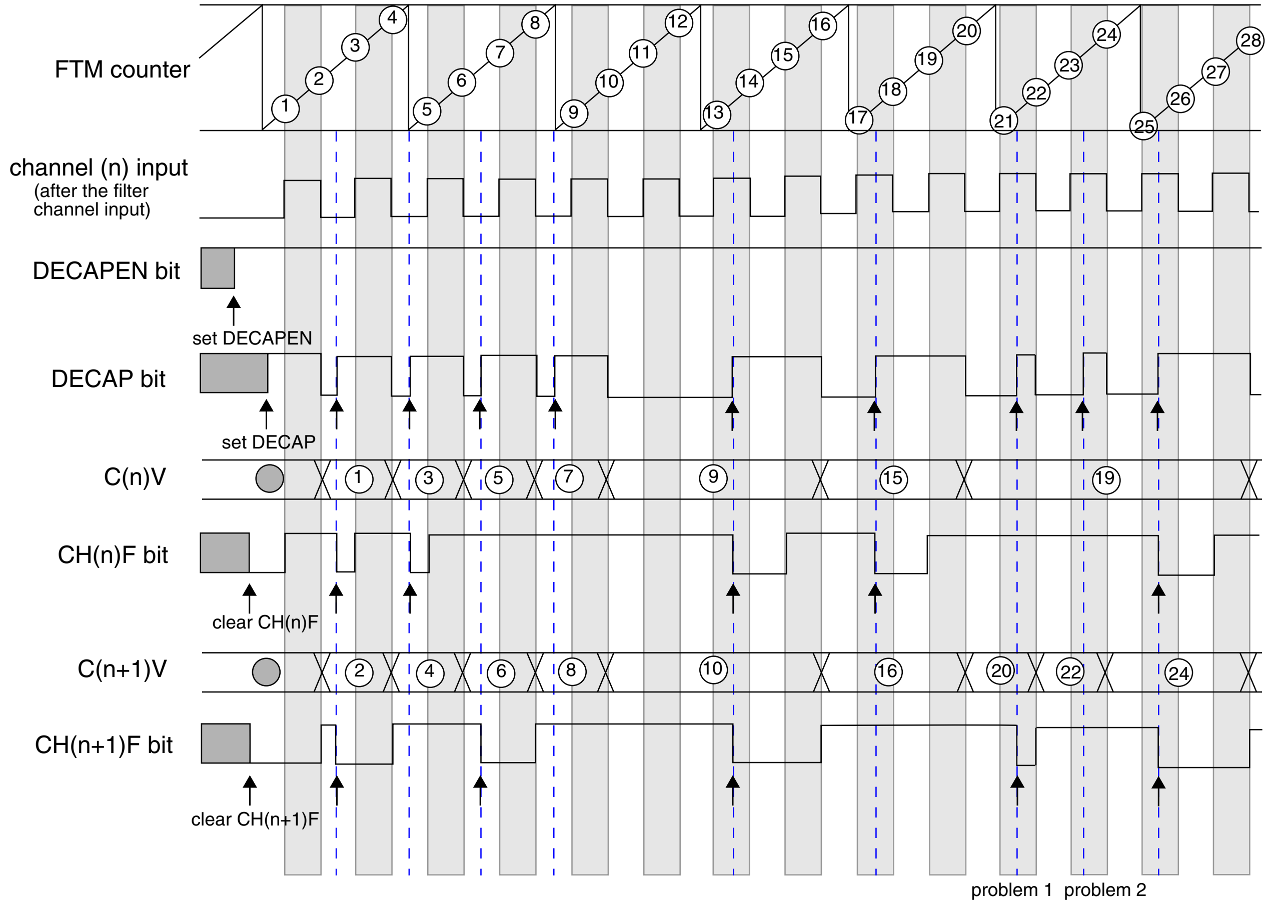
For a new sequence of the measurements in the Dual Edge Capture – Continuous mode, clear the CH(n)F and CH(n+1)F bits to start new measurements.

39.4.24.3 Pulse width measurement

If the channel (n) is configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1) and the channel (n+1) to capture falling edges (ELS(n+1)B:ELS(n+1)A = 1:0), then the positive polarity pulse width is measured. If the channel (n) is configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0) and the channel (n+1) to capture rising edges (ELS(n+1)B:ELS(n+1)A = 0:1), then the negative polarity pulse width is measured.

The pulse width measurement can be made in One-Shot Capture mode or Continuous Capture mode.

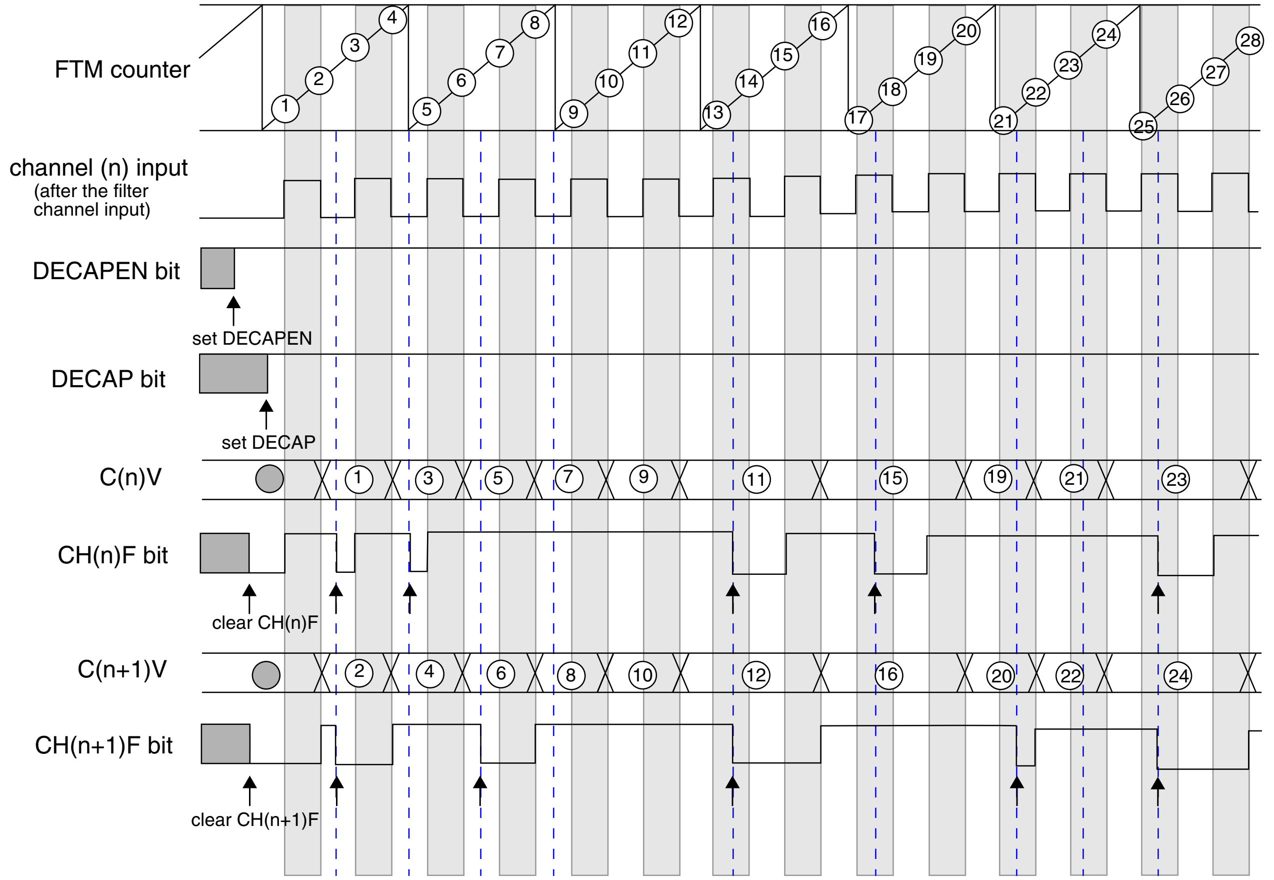
The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next positive polarity pulse width. The CH(n)F bit is set when the first edge of this pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second edge of this pulse is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 39-286. Dual Edge Capture – One-Shot mode for positive polarity pulse width measurement

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first edge of the positive polarity pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set when the second edge of this pulse is detected, that is, the edge selected by ELS(n +1)B:ELS(n+1)A bits. The CH(n+1)F bit indicates when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

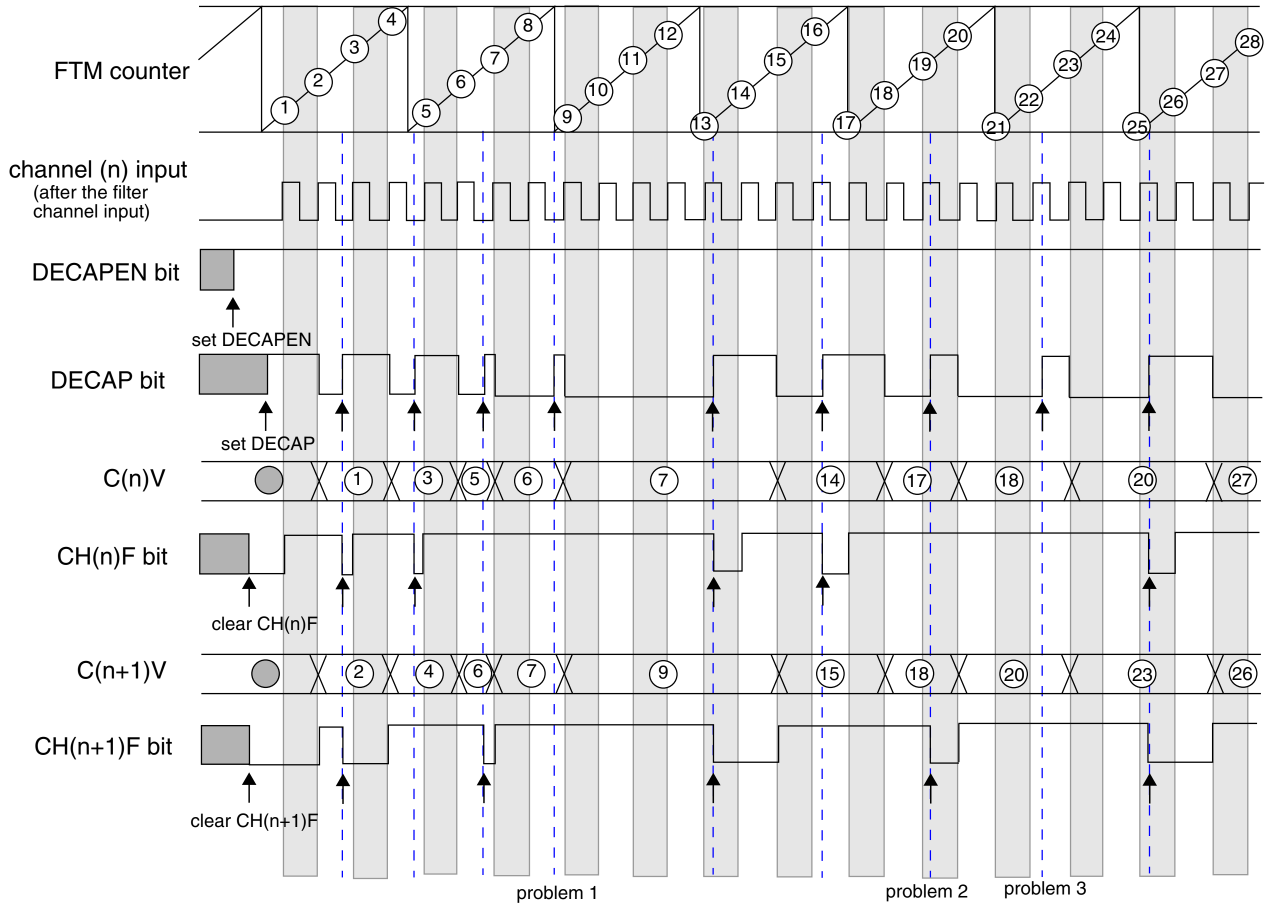
Figure 39-287. Dual Edge Capture – Continuous mode for positive polarity pulse width measurement

39.4.24.4 Period measurement

If the channels (n) and (n+1) are configured to capture consecutive edges of the same polarity, then the period of the channel (n) input signal is measured. If both channels (n) and (n+1) are configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1 and ELS(n+1)B:ELS(n+1)A = 0:1), then the period between two consecutive rising edges is measured. If both channels (n) and (n+1) are configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0 and ELS(n+1)B:ELS(n+1)A = 1:0), then the period between two consecutive falling edges is measured.

The period measurement can be made in One-Shot Capture mode or Continuous Capture mode.

The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next period. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second rising edge is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two selected edges were captured and the C(n)V and C(n+1)V registers are ready for reading.

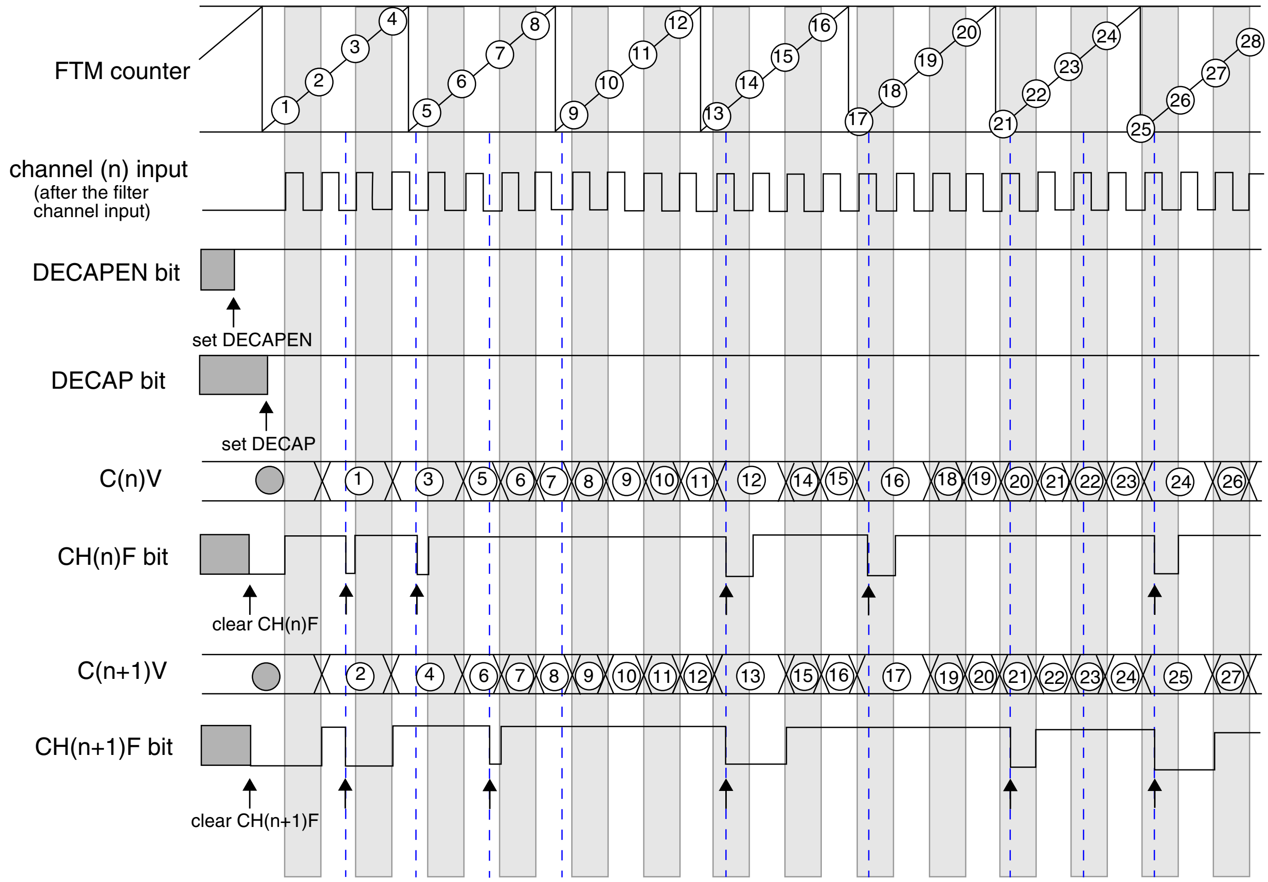


Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 0, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 3: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 39-288. Dual Edge Capture – One-Shot mode to measure of the period between two consecutive rising edges

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set

when the second rising edge is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. The CH(n+1)F bit indicates when two edges of the period were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

Figure 39-289. Dual Edge Capture – Continuous mode to measure of the period between two consecutive rising edges

39.4.24.5 Read coherency mechanism

The Dual Edge Capture mode implements a read coherency mechanism between the FTM counter value captured in C(n)V and C(n+1)V registers. The read coherency mechanism is illustrated in the following figure. In this example, the channels (n) and (n+1) are in Dual Edge Capture – Continuous mode for positive polarity pulse width measurement. Thus, the channel (n) is configured to capture the FTM counter value when there is a rising edge at channel (n) input signal, and channel (n+1) to capture the FTM counter value when there is a falling edge at channel (n) input signal.

When a rising edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n) capture buffer. The channel (n) capture buffer value is transferred to C(n)V register when a falling edge occurs in the channel (n) input signal. C(n)V register has the FTM counter value when the previous rising edge occurred, and the channel (n) capture buffer has the FTM counter value when the last rising edge occurred.

When a falling edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n+1) capture buffer. The channel (n+1) capture buffer value is transferred to C(n+1)V register when the C(n)V register is read.

In the following figure, the read of C(n)V returns the FTM counter value when the event 1 occurred and the read of C(n+1)V returns the FTM counter value when the event 2 occurred.

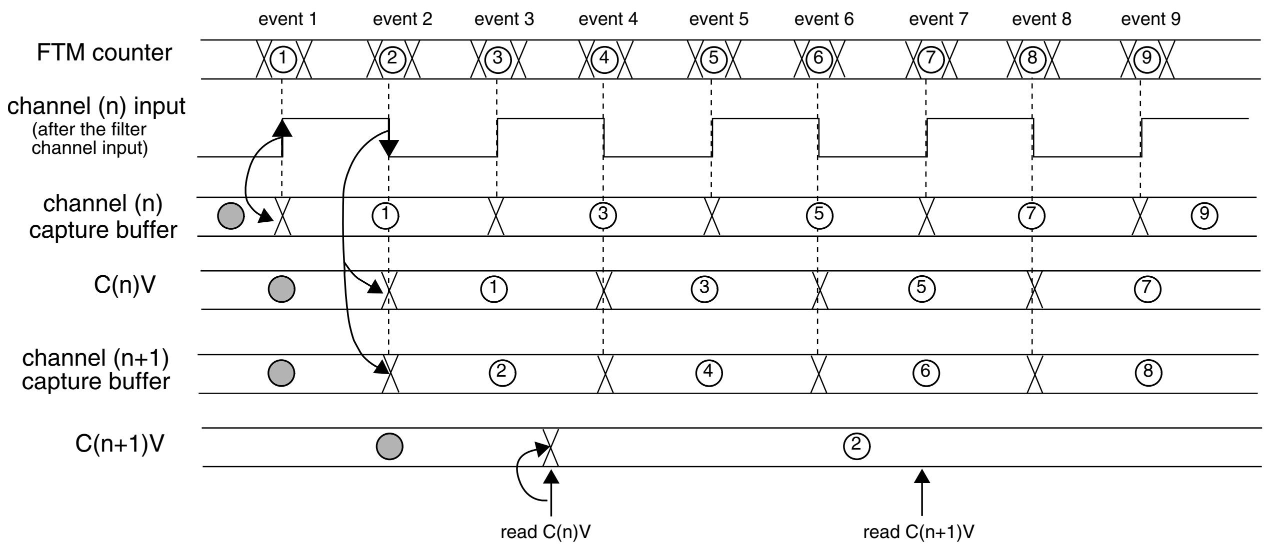


Figure 39-290. Dual Edge Capture mode read coherency mechanism

C(n)V register must be read prior to C(n+1)V register in dual edge capture one-shot and continuous modes for the read coherency mechanism works properly.

39.4.25 Quadrature Decoder mode

The Quadrature Decoder mode is selected if (FTMEN = 1) and (QUADEN = 1). The Quadrature Decoder mode uses the input signals phase A and B to control the FTM counter increment and decrement. The following figure shows the quadrature decoder block diagram.

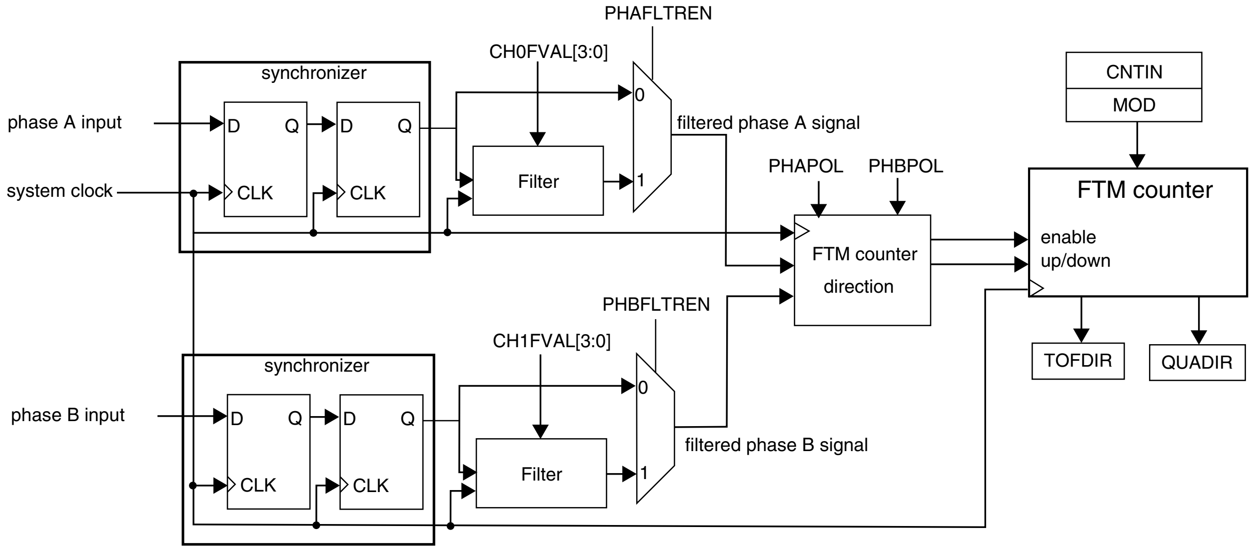


Figure 39-291. Quadrature Decoder block diagram

Each one of input signals phase A and B has a filter that is equivalent to the filter used in the channels input; Filter for Input Capture mode. The phase A input filter is enabled by PHAFLTREN bit and this filter's value is defined by CH0FVAL[3:0] bits (CH(n)FVAL[3:0] bits in FILTER0 register). The phase B input filter is enabled by PHBFLTREN bit and this filter's value is defined by CH1FVAL[3:0] bits (CH(n +1)FVAL[3:0] bits in FILTER0 register).

Except for CH0FVAL[3:0] and CH1FVAL[3:0] bits, no channel logic is used in Quadrature Decoder mode.

Note

Notice that the FTM counter is clocked by the phase A and B input signals when quadrature decoder mode is selected. Therefore it is expected that the Quadrature Decoder be used only with the FTM channels in input capture or output compare modes.

The PHAPOL bit selects the polarity of the phase A input, and the PHBPOL bit selects the polarity of the phase B input.

The QUADMODE selects the encoding mode used in the Quadrature Decoder mode. If QUADMODE = 1, then the count and direction encoding mode is enabled; see the following figure. In this mode, the phase B input value indicates the counting direction, and the phase A input defines the counting rate. The FTM counter is updated when there is a rising edge at phase A input signal.

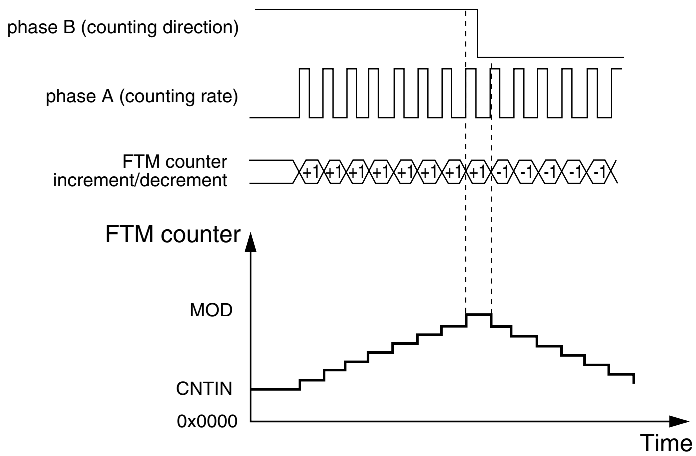


Figure 39-292. Quadrature Decoder – Count and Direction Encoding mode

If QUADMODE = 0, then the Phase A and Phase B Encoding mode is enabled; see the following figure. In this mode, the relationship between phase A and B signals indicates the counting direction, and phase A and B signals define the counting rate. The FTM counter is updated when there is an edge either at the phase A or phase B signals.

If PHAPOL = 0 and PHBPOL = 0, then the FTM counter increment happens when:

- there is a rising edge at phase A signal and phase B signal is at logic zero;
- there is a rising edge at phase B signal and phase A signal is at logic one;
- there is a falling edge at phase B signal and phase A signal is at logic zero;
- there is a falling edge at phase A signal and phase B signal is at logic one;

and the FTM counter decrement happens when:

- there is a falling edge at phase A signal and phase B signal is at logic zero;
- there is a falling edge at phase B signal and phase A signal is at logic one;
- there is a rising edge at phase B signal and phase A signal is at logic zero;
- there is a rising edge at phase A signal and phase B signal is at logic one.

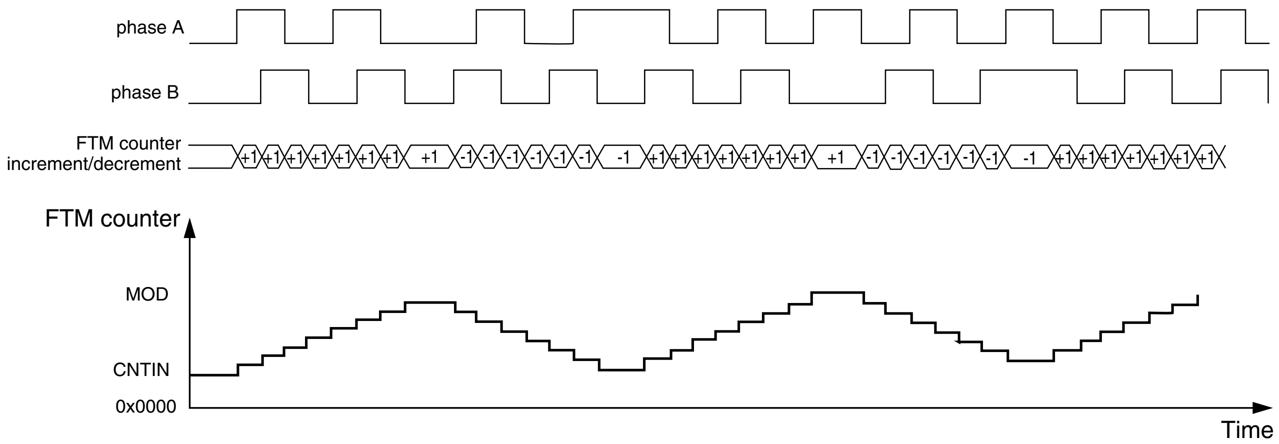


Figure 39-293. Quadrature Decoder – Phase A and Phase B Encoding mode

The following figure shows the FTM counter overflow in up counting. In this case, when the FTM counter changes from MOD to CNTIN, TOF and TOFDIR bits are set. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was up when the FTM counter overflow occurred.

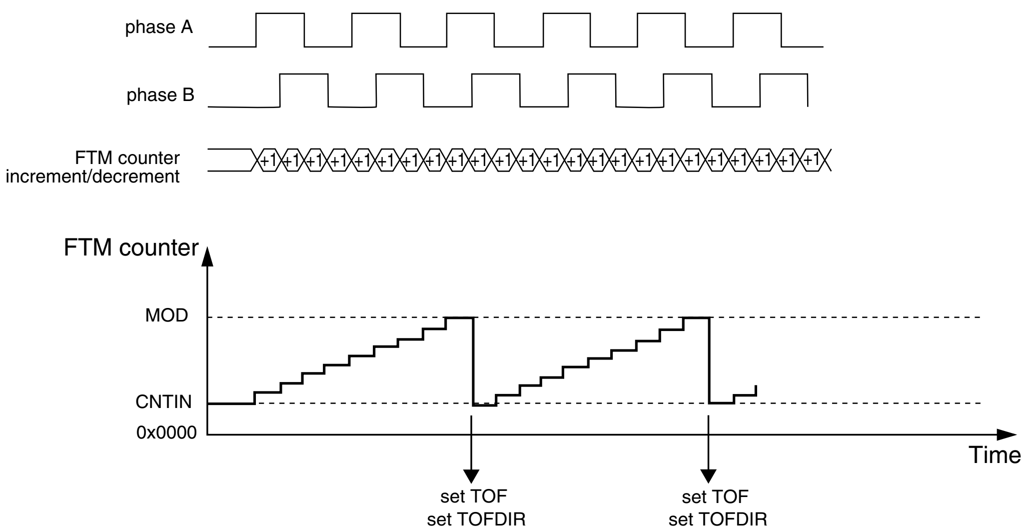


Figure 39-294. FTM Counter overflow in up counting for Quadrature Decoder mode

The following figure shows the FTM counter overflow in down counting. In this case, when the FTM counter changes from CNTIN to MOD, TOF bit is set and TOFDIR bit is cleared. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was down when the FTM counter overflow occurred.

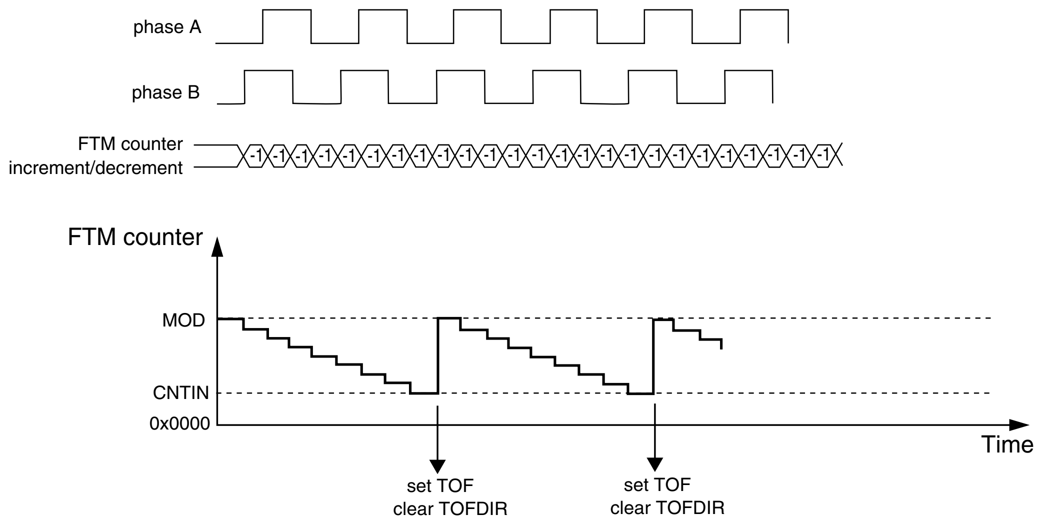


Figure 39-295. FTM counter overflow in down counting for Quadrature Decoder mode

39.4.25.1 Quadrature Decoder boundary conditions

The following figures show the FTM counter responding to motor jittering typical in motor position control applications.

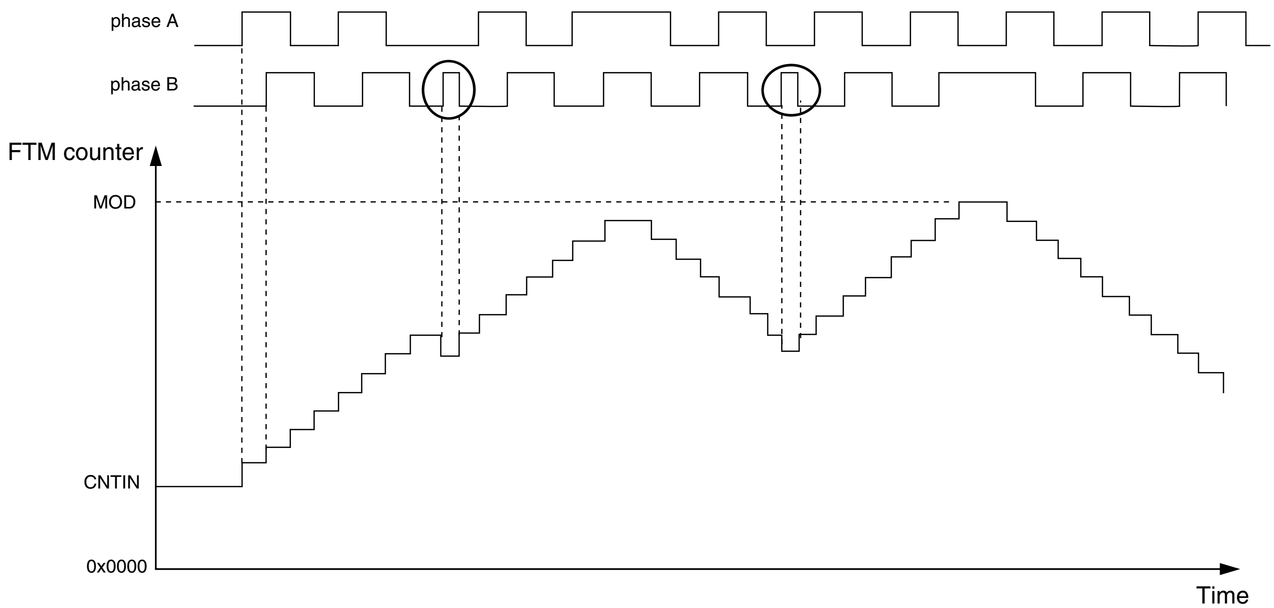


Figure 39-296. Motor position jittering in a mid count value

The following figure shows motor jittering produced by the phase B and A pulses respectively:

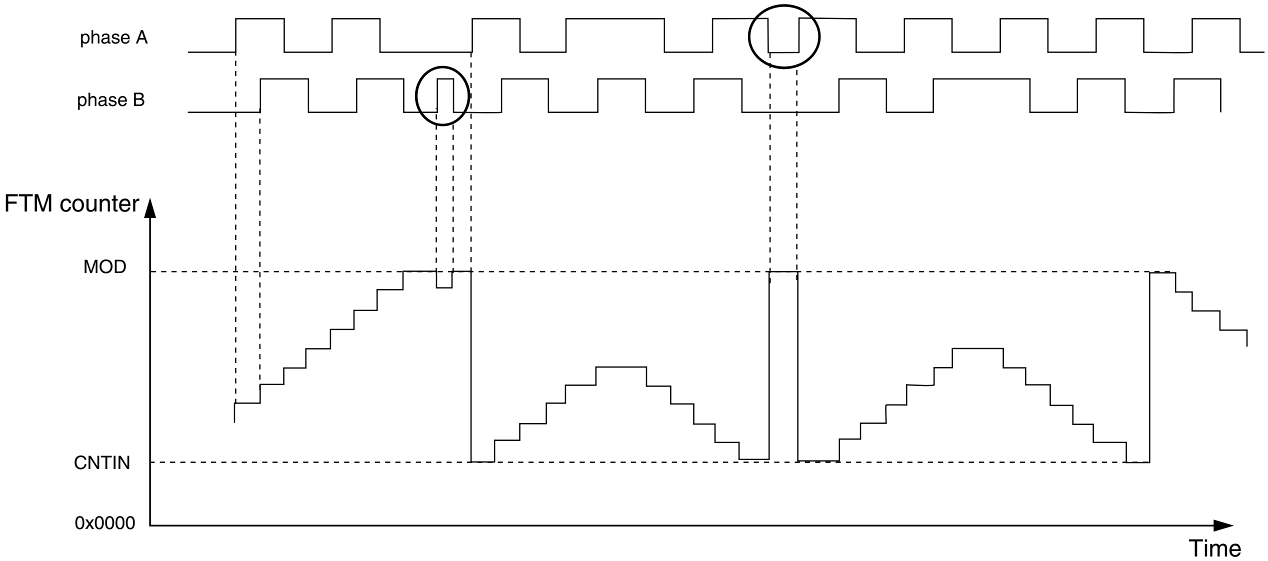


Figure 39-297. Motor position jittering near maximum and minimum count value

The first highlighted transition causes a jitter on the FTM counter value near the maximum count value (MOD). The second indicated transition occurs on phase A and causes the FTM counter transition between the maximum and minimum count values which are defined by MOD and CNTIN registers.

The appropriate settings of the phase A and phase B input filters are important to avoid glitches that may cause oscillation on the FTM counter value. The preceding figures show examples of oscillations that can be caused by poor input filter setup. Thus, it is important to guarantee a minimum pulse width to avoid these oscillations.

39.4.26 BDM mode

When the chip is in BDM mode, the BDMMODE[1:0] bits select the behavior of the FTM counter, the CH(n)F bit, the channels output, and the writes to the MOD, CNTIN, and C(n)V registers according to the following table.

Table 39-313. FTM behavior when the chip Is in BDM mode

BDMMODE	FTM Counter	CH(n)F Bit	FTM Channels Output	Writes to MOD, CNTIN, and C(n)V Registers
00	Stopped	can be set	Functional mode	Writes to these registers bypass the registers buffers
01	Stopped	is not set	The channels outputs are forced to their safe value according to POLn bit	Writes to these registers bypass the registers buffers
10	Stopped	is not set	The channels outputs are frozen when the chip enters in BDM mode	Writes to these registers bypass the registers buffers

Table continues on the next page...

Table 39-313. FTM behavior when the chip Is in BDM mode (continued)

BDMMODE	FTM Counter	CH(n)F Bit	FTM Channels Output	Writes to MOD, CNTIN, and C(n)V Registers
11	Functional mode	can be set	Functional mode	Functional mode

Note that if BDMMODE[1:0] = 2'b00 then the channels outputs remain at the value when the chip enters in BDM mode, because the FTM counter is stopped. However, the following situations modify the channels outputs in this BDM mode.

- Write any value to CNT register; see Counter reset. In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for those channels set to Output Compare mode.
- FTM counter is reset by PWM Synchronization mode; see FTM counter synchronization. In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for channels in Output Compare mode.
- In the channels outputs initialization, the channel (n) output is forced to the CH(n)OI bit value when the value 1 is written to INIT bit. See Initialization.

Note

The BDMMODE[1:0] = 2'b00 must not be used with the Fault control. Even if the fault control is enabled and a fault condition exists, the channels outputs values are updated as above.

39.4.27 Intermediate load

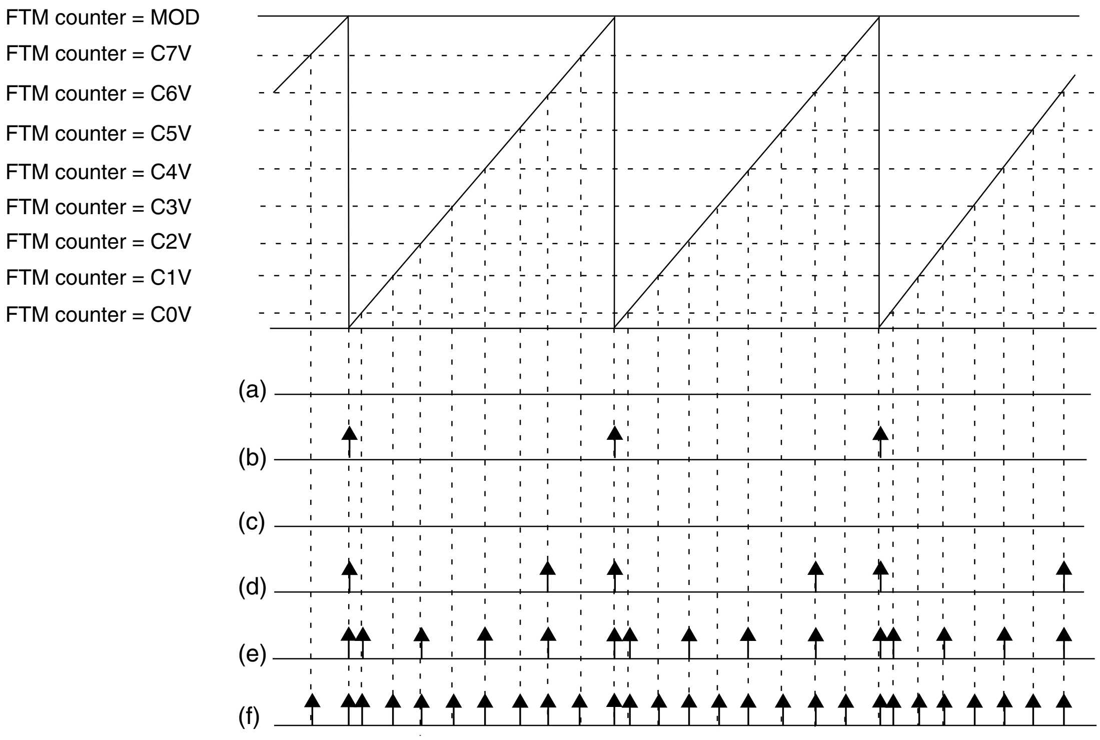
The PWMLOAD register allows to update the MOD, CNTIN, and C(n)V registers with the content of the register buffer at a defined load point. In this case, it is not required to use the PWM synchronization.

There are multiple possible loading points for intermediate load:

Table 39-314. When possible loading points are enabled

Loading point	Enabled
When the FTM counter wraps from MOD value to CNTIN value	Always
At the channel (j) match (FTM counter = C(j)V)	When CHjSEL = 1

The following figure shows some examples of enabled loading points.



NOTE

(a) LDOK = 0, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0
(b) LDOK = 1, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0
(c) LDOK = 0, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 1, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0
(d) LDOK = 1, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 1, CH7SEL = 0
(e) LDOK = 1, CH0SEL = 1, CH1SEL = 0, CH2SEL = 1, CH3SEL = 0, CH4SEL = 1, CH5SEL = 0, CH6SEL = 1, CH7SEL = 0
(f) LDOK = 1, CH0SEL = 1, CH1SEL = 1, CH2SEL = 1, CH3SEL = 1, CH4SEL = 1, CH5SEL = 1, CH6SEL = 1, CH7SEL = 1

Figure 39-298. Loading points for intermediate load

After enabling the loading points, the LDOK bit must be set for the load to occur. In this case, the load occurs at the next enabled loading point according to the following conditions:

Table 39-315. Conditions for loads occurring at the next enabled loading point

When a new value was written	Then
To the MOD register	The MOD register is updated with its write buffer value.
To the CNTIN register and CNTINC = 1	The CNTIN register is updated with its write buffer value.
To the C(n)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n)V register is updated with its write buffer value.
To the C(n+1)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n+1)V register is updated with its write buffer value.

NOTE

- If ELSjB and ELSjA bits are different from zero, then the channel (j) output signal is generated according to the configured output mode. If ELSjB and ELSjA bits are zero, then the generated signal is not available on channel (j) output.
- If CHjIE = 1, then the channel (j) interrupt is generated when the channel (j) match occurs.
- At the intermediate load neither the channels outputs nor the FTM counter are changed. Software must set the intermediate load at a safe point in time.
- The intermediate load feature must be used only in Combine mode.

39.4.28 Global time base (GTB)

The global time base (GTB) is a FTM function that allows the synchronization of multiple FTM modules on a chip. The following figure shows an example of the GTB feature used to synchronize two FTM modules. In this case, the FTM A and B channels can behave as if just one FTM module was used, that is, a global time base.

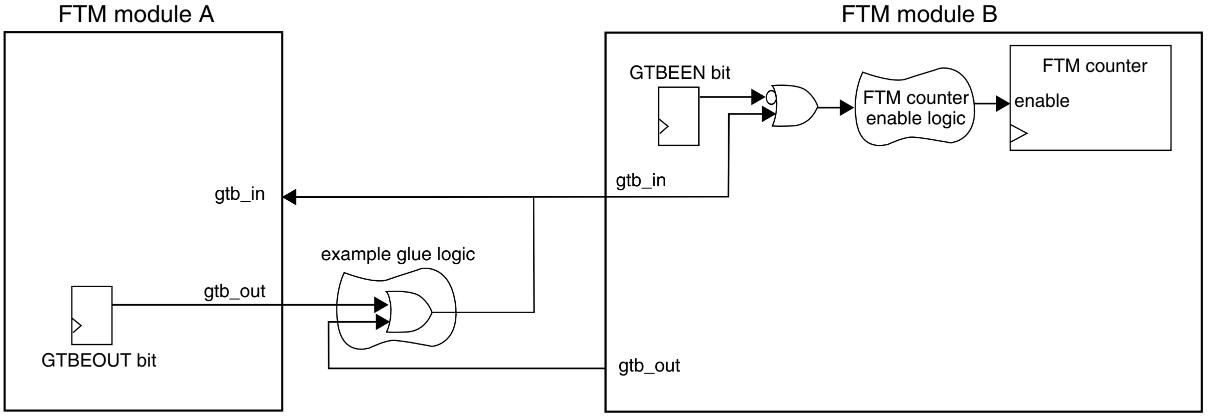


Figure 39-299. Global time base (GTB) block diagram

The GTB functionality is implemented by the GTBEEN and GTBEOUT bits in the CONF register, the input signal *gtb_in*, and the output signal *gtb_out*. The GTBEEN bit enables gtb_in to control the FTM counter enable signal:

- If GTBEEN = 0, each one of FTM modules works independently according to their configured mode.
- If GTBEEN = 1, the FTM counter update is enabled only when gtb_in is 1.

In the configuration described in the preceding figure, FTM modules A and B have their FTM counters enabled if at least one of the gtb_out signals from one of the FTM modules is 1. There are several possible configurations for the interconnection of the gtb_in and gtb_out signals, represented by the example glue logic shown in the figure. Note that these configurations are chip-dependent and implemented outside of the FTM modules. See the chip configuration details for the chip's specific implementation.

NOTE

- In order to use the GTB signals to synchronize the FTM counter of different FTM modules, the configuration of each FTM module should guarantee that its FTM counter starts counting as soon as the gtb_in signal is 1.
- The GTB feature does not provide continuous synchronization of FTM counters, meaning that the FTM counters may lose synchronization during FTM operation. The GTB feature only allows the FTM counters to *start* their operation synchronously.

39.4.28.1 Enabling the global time base (GTB)

To enable the GTB feature, follow these steps for each participating FTM module:

1. Stop the FTM counter: Write 00b to SC[CLKS].
2. Program the FTM to the intended configuration. The FTM counter mode needs to be consistent across all participating modules.
3. Write 1 to CONF[GTBEEN] and write 0 to CONF[GTBEOUT] at the same time.
4. Select the intended FTM counter clock source in SC[CLKS]. The clock source needs to be consistent across all participating modules.
5. Reset the FTM counter: Write any value to the CNT register.

To initiate the GTB feature in the configuration described in the preceding figure, write 1 to CONF[GTBEOUT] in the FTM module used as the time base.

39.5 Reset overview

The FTM is reset whenever any chip reset occurs.

When the FTM exits from reset:

- the FTM counter and the prescaler counter are zero and are stopped (CLKS[1:0] = 00b);
- the timer overflow interrupt is zero, see Timer Overflow Interrupt;

- the channels interrupts are zero, see Channel (n) Interrupt;
- the fault interrupt is zero, see Fault Interrupt;
- the channels are in input capture mode, see Input Capture mode;
- the channels outputs are zero;
- the channels pins are not controlled by FTM (ELS(n)B:ELS(n)A = 0:0) (See the table in the description of CnSC register).

The following figure shows the FTM behavior after the reset. At the reset (item 1), the FTM counter is disabled (see the description of the CLKS field in the Status and Control register), its value is updated to zero and the pins are not controlled by FTM (See the table in the description of CnSC register).

After the reset, the FTM should be configurated (item 2). It is necessary to define the FTM counter mode, the FTM counting limits (MOD and CNTIN registers value), the channels mode and CnV registers value according to the channels mode.

Thus, it is recommended to write any value to CNT register (item 3). This write updates the FTM counter with the CNTIN register value and the channels output with its initial value (except for channels in output compare mode) (Counter reset).

The next step is to select the FTM counter clock by the CLKS[1:0] bits (item 4). It is important to highlight that the pins are only controlled by FTM when CLKS[1:0] bits are different from zero (See the table in the description of CnSC register).

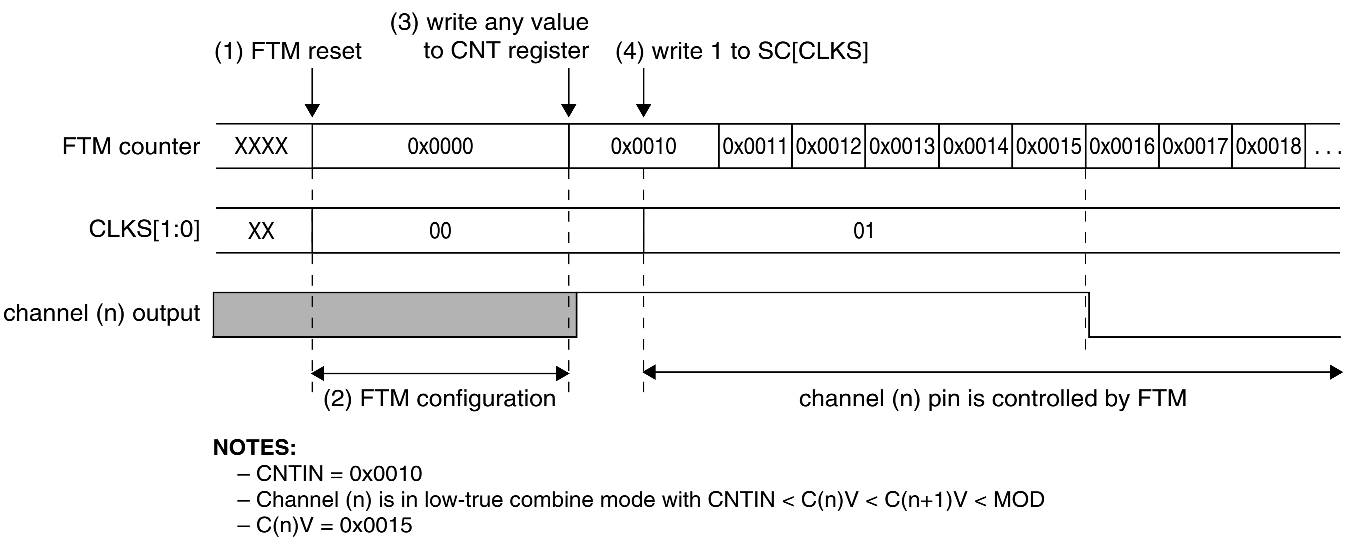


Figure 39-300. FTM behavior after reset when the channel (n) is in Combine mode

The following figure shows an example when the channel (n) is in Output Compare mode and the channel (n) output is toggled when there is a match. In the Output Compare mode, the channel output is not updated to its initial value when there is a write to CNT

register (item 3). In this case, use the software output control (Software output control) or the initialization (Initialization) to update the channel output to the selected value (item 4).

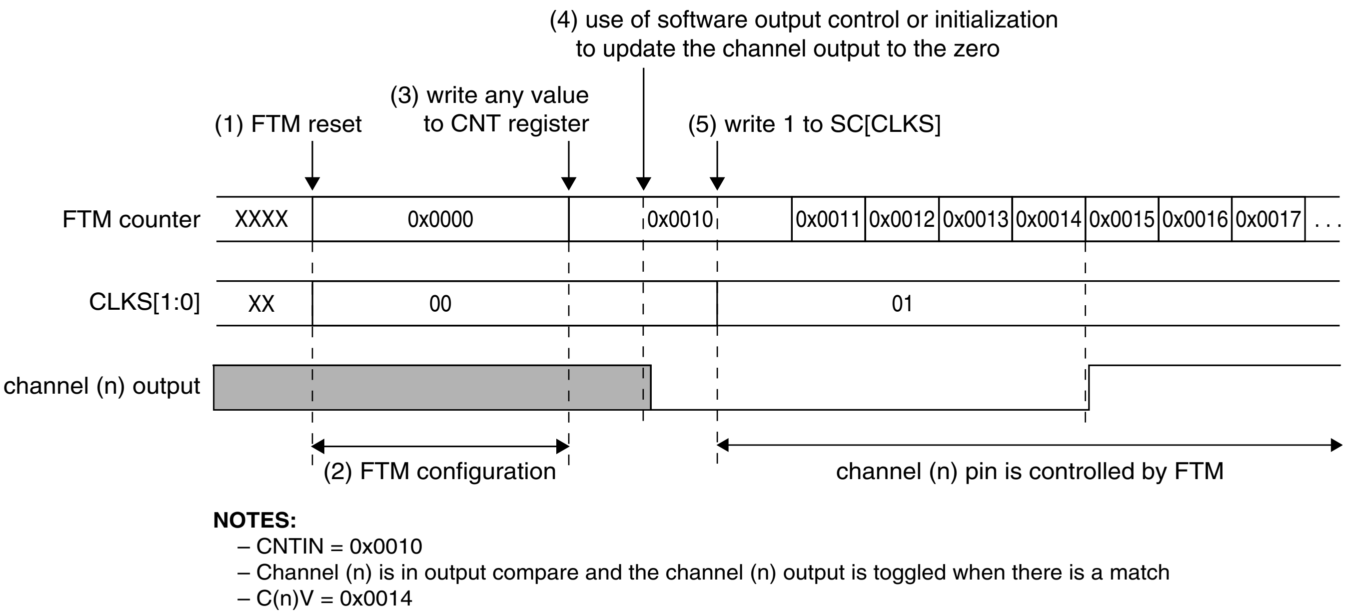


Figure 39-301. FTM behavior after reset when the channel (n) is in Output Compare mode

39.6 FTM Interrupts

39.6.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

39.6.2 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHnIE = 1) and (CHnF = 1).

39.6.3 Fault Interrupt

The fault interrupt is generated when (FAULTIE = 1) and (FAULTF = 1).

Chapter 40
Periodic Interrupt Timer (PIT)

40.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The PIT module is an array of timers that can be used to raise interrupts and trigger DMA channels.

40.1.1 Block diagram

The following figure shows the block diagram of the PIT module.

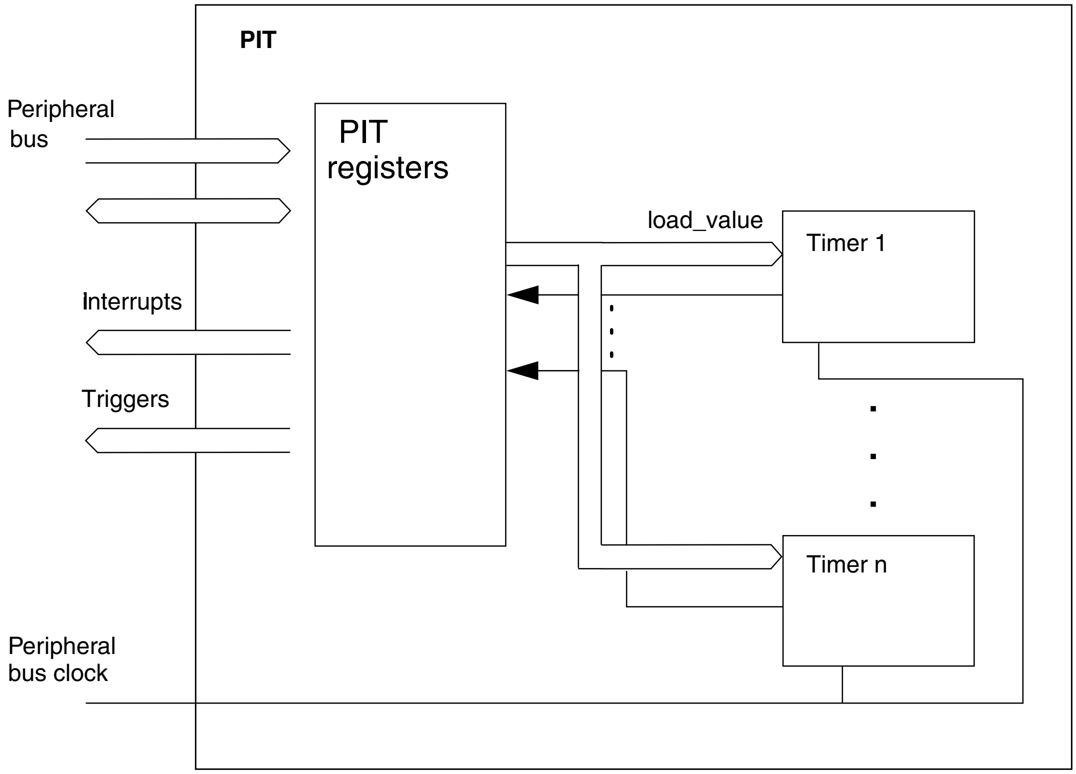


Figure 40-1. Block diagram of the PIT

NOTE

See the chip configuration details for the number of PIT channels used in this MCU.

40.1.2 Features

The main features of this block are:

- Ability of timers to generate DMA trigger pulses
- Ability of timers to generate interrupts
- Maskable interrupts
- Independent timeout periods for each timer

40.2 Signal description

The PIT module has no external pins.

40.3 Memory map/register description

This section provides a detailed description of all registers accessible in the PIT module.

- Reserved registers will read as 0, writes will have no effect.
- See the chip configuration details for the number of PIT channels used in this MCU.

PIT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_7000	PIT Module Control Register (PIT_MCR)	32	R/W	0000_0006h	40.3.1/1876
4003_70E0	PIT Upper Lifetime Timer Register (PIT_LTMR64H)	32	R	0000_0000h	40.3.2/1877
4003_70E4	PIT Lower Lifetime Timer Register (PIT_LTMR64L)	32	R	0000_0000h	40.3.3/1877
4003_7100	Timer Load Value Register (PIT_LDVAL0)	32	R/W	0000_0000h	40.3.4/1878
4003_7104	Current Timer Value Register (PIT_CVAL0)	32	R	0000_0000h	40.3.5/1879
4003_7108	Timer Control Register (PIT_TCTRL0)	32	R/W	0000_0000h	40.3.6/1879
4003_710C	Timer Flag Register (PIT_TFLG0)	32	R/W	0000_0000h	40.3.7/1880
4003_7110	Timer Load Value Register (PIT_LDVAL1)	32	R/W	0000_0000h	40.3.4/1878
4003_7114	Current Timer Value Register (PIT_CVAL1)	32	R	0000_0000h	40.3.5/1879
4003_7118	Timer Control Register (PIT_TCTRL1)	32	R/W	0000_0000h	40.3.6/1879
4003_711C	Timer Flag Register (PIT_TFLG1)	32	R/W	0000_0000h	40.3.7/1880
4003_7120	Timer Load Value Register (PIT_LDVAL2)	32	R/W	0000_0000h	40.3.4/1878
4003_7124	Current Timer Value Register (PIT_CVAL2)	32	R	0000_0000h	40.3.5/1879
4003_7128	Timer Control Register (PIT_TCTRL2)	32	R/W	0000_0000h	40.3.6/1879
4003_712C	Timer Flag Register (PIT_TFLG2)	32	R/W	0000_0000h	40.3.7/1880
4003_7130	Timer Load Value Register (PIT_LDVAL3)	32	R/W	0000_0000h	40.3.4/1878
4003_7134	Current Timer Value Register (PIT_CVAL3)	32	R	0000_0000h	40.3.5/1879
4003_7138	Timer Control Register (PIT_TCTRL3)	32	R/W	0000_0000h	40.3.6/1879
4003_713C	Timer Flag Register (PIT_TFLG3)	32	R/W	0000_0000h	40.3.7/1880
4003_7140	Timer Load Value Register (PIT_LDVAL4)	32	R/W	0000_0000h	40.3.4/1878
4003_7144	Current Timer Value Register (PIT_CVAL4)	32	R	0000_0000h	40.3.5/1879
4003_7148	Timer Control Register (PIT_TCTRL4)	32	R/W	0000_0000h	40.3.6/1879
4003_714C	Timer Flag Register (PIT_TFLG4)	32	R/W	0000_0000h	40.3.7/1880
4003_7150	Timer Load Value Register (PIT_LDVAL5)	32	R/W	0000_0000h	40.3.4/1878
4003_7154	Current Timer Value Register (PIT_CVAL5)	32	R	0000_0000h	40.3.5/1879
4003_7158	Timer Control Register (PIT_TCTRL5)	32	R/W	0000_0000h	40.3.6/1879
4003_715C	Timer Flag Register (PIT_TFLG5)	32	R/W	0000_0000h	40.3.7/1880
4003_7160	Timer Load Value Register (PIT_LDVAL6)	32	R/W	0000_0000h	40.3.4/1878
4003_7164	Current Timer Value Register (PIT_CVAL6)	32	R	0000_0000h	40.3.5/1879
4003_7168	Timer Control Register (PIT_TCTRL6)	32	R/W	0000_0000h	40.3.6/1879

Table continues on the next page...

PIT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_716C	Timer Flag Register (PIT_TFLG6)	32	R/W	0000_0000h	40.3.7/1880
4003_7170	Timer Load Value Register (PIT_LDVAL7)	32	R/W	0000_0000h	40.3.4/1878
4003_7174	Current Timer Value Register (PIT_CVAL7)	32	R	0000_0000h	40.3.5/1879
4003_7178	Timer Control Register (PIT_TCTRL7)	32	R/W	0000_0000h	40.3.6/1879
4003_717C	Timer Flag Register (PIT_TFLG7)	32	R/W	0000_0000h	40.3.7/1880

40.3.1 PIT Module Control Register (PIT_MCR)

This register enables or disables the PIT timer clocks and controls the timers when the PIT enters the Debug mode.

Address: 4003_7000h base + 0h offset = 4003_7000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													Reserved	MDIS	FRZ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

PIT_MCR field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved.
1 MDIS	Module Disable - (PIT section) Disables the standard timers. This field must be enabled before any other setup is done. 0 Clock for standard PIT timers is enabled. 1 Clock for standard PIT timers is disabled.
0 FRZ	Freeze Allows the timers to be stopped when the device enters the Debug mode. 0 Timers continue to run in Debug mode. 1 Timers are stopped in Debug mode.

40.3.2 PIT Upper Lifetime Timer Register (PIT_LTMR64H)

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit lifetimer.

Address: 4003_7000h base + E0h offset = 4003_70E0h

Bit	31–0
R	LTH
W	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

PIT_LTMR64H field descriptions

Field	Description
31–0 LTH	Life Timer value Shows the timer value of timer 1. If this register is read at a time t1, LTMR64L shows the value of timer 0 at time t1.

40.3.3 PIT Lower Lifetime Timer Register (PIT_LTMR64L)

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit lifetimer.

To use LTMR64H and LTMR64L, timer 0 and timer 1 need to be chained. To obtain the correct value, first read LTMR64H and then LTMR64L. LTMR64H will have the value of CVAL1 at the time of the first access, LTMR64L will have the value of CVAL0 at the time of the first access, therefore the application does not need to worry about carry-over effects of the running counter.

Address: 4003_7000h base + E4h offset = 4003_70E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LTL																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_LTMR64L field descriptions

Field	Description
31–0 LTL	Life Timer value Shows the value of timer 0 at the time LTMR64H was last read. It will only update if LTMR64H is read.

40.3.4 Timer Load Value Register (PIT_LDVAL*n*)

These registers select the timeout period for the timer interrupts.

Address: 4003_7000h base + 100h offset + (16d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSV																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_LDVAL*n* field descriptions

Field	Description
31–0 TSV	Timer Start Value Sets the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer; instead the value will be loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again.

40.3.5 Current Timer Value Register (PIT_CVAL*n*)

These registers indicate the current timer position.

Address: 4003_7000h base + 104h offset + (16d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TVL																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_CVAL*n* field descriptions

Field	Description
31–0 TVL	Current Timer Value Represents the current timer value, if the timer is enabled. **NOTE:** • If the timer is disabled, do not use this field as its value is unreliable. • The timer uses a downcounter. The timer values are frozen in Debug mode if MCR[FRZ] is set.

40.3.6 Timer Control Register (PIT_TCTRL*n*)

These registers contain the control bits for each timer.

Address: 4003_7000h base + 108h offset + (16d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													CHN	TIE	TEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TCTRL*n* field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CHN	Chain Mode When activated, Timer n-1 needs to expire before timer n can decrement by 1. Timer 0 cannot be chained.

Table continues on the next page...

PIT_TCTRLn field descriptions (continued)

Field	Description
	0 Timer is not chained. 1 Timer is chained to previous timer. For example, for Channel 2, if this field is set, Timer 2 is chained to Timer 1.
1 TIE	Timer Interrupt Enable When an interrupt is pending, or, TFLGn[TIF] is set, enabling the interrupt will immediately cause an interrupt event. To avoid this, the associated TFLGn[TIF] must be cleared first. 0 Interrupt requests from Timer n are disabled. 1 Interrupt will be requested whenever TIF is set.
0 TEN	Timer Enable Enables or disables the timer. 0 Timer n is disabled. 1 Timer n is enabled.

40.3.7 Timer Flag Register (PIT_TFLG*n*)

These registers hold the PIT interrupt flags.

Address: 4003_7000h base + 10Ch offset + (16d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															TIF
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TFLGn field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 TIF	Timer Interrupt Flag Sets to 1 at the end of the timer period. Writing 1 to this flag clears it. Writing 0 has no effect. If enabled, or, when TCTRLn[TIE] = 1, TIF causes an interrupt request. 0 Timeout has not yet occurred. 1 Timeout has occurred.

40.4 Functional description

This section provides the functional description of the module.

40.4.1 General operation

This section gives detailed information on the internal operation of the module. Each timer can be used to generate trigger pulses and interrupts. Each interrupt is available on a separate interrupt line.

40.4.1.1 Timers

The timers generate triggers at periodic intervals, when enabled. The timers load the start values as specified in their LDVAL registers, count down to 0 and then load the respective start value again. Each time a timer reaches 0, it will generate a trigger pulse and set the interrupt flag.

All interrupts can be enabled or masked by setting TCTRLn[TIE]. A new interrupt can be generated only after the previous one is cleared.

If desired, the current counter value of the timer can be read via the CVAL registers.

The counter period can be restarted, by first disabling, and then enabling the timer with TCTRLn[TEN]. See the following figure.

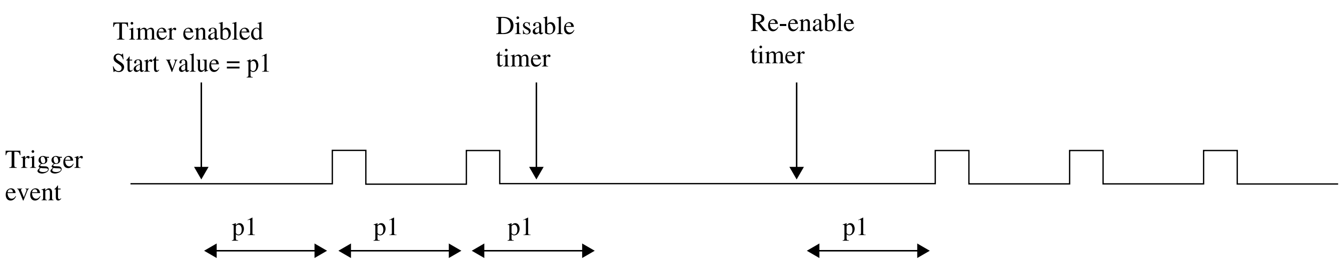


Figure 40-41. Stopping and starting a timer

The counter period of a running timer can be modified, by first disabling the timer, setting a new load value, and then enabling the timer again. See the following figure.

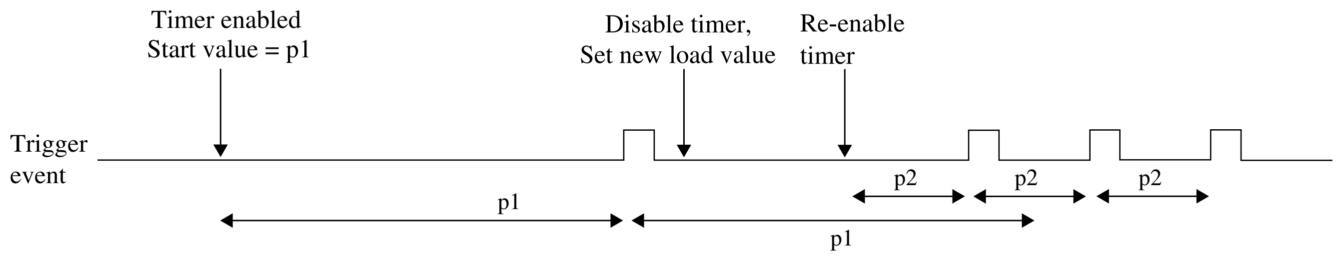


Figure 40-42. Modifying running timer period

It is also possible to change the counter period without restarting the timer by writing LDVAL with the new load value. This value will then be loaded after the next trigger event. See the following figure.

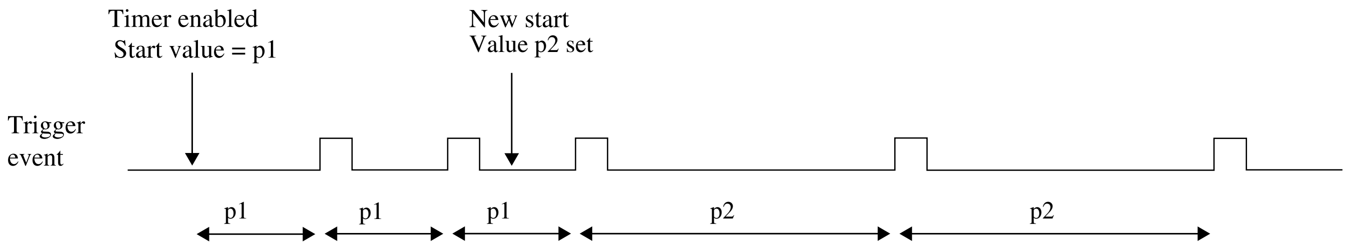


Figure 40-43. Dynamically setting a new load value

40.4.1.2 Debug mode

In Debug mode, the timers will be frozen based on MCR[FRZ]. This is intended to aid software development, allowing the developer to halt the processor, investigate the current state of the system, for example, the timer values, and then continue the operation.

40.4.2 Interrupts

All the timers support interrupt generation. See the MCU specification for related vector addresses and priorities.

Timer interrupts can be enabled by setting TCTRLn[TIE]. TFLGn[TIF] are set to 1 when a timeout occurs on the associated timer, and are cleared to 0 by writing a 1 to the corresponding TFLGn[TIF].

40.4.3 Chained timers

When a timer has chain mode enabled, it will only count after the previous timer has expired. So if timer n-1 has counted down to 0, counter n will decrement the value by one. This allows to chain some of the timers together to form a longer timer. The first timer (timer 0) cannot be chained to any other timer.

40.5 Initialization and application information

In the example configuration:

- The PIT clock has a frequency of 50 MHz.
- Timer 1 creates an interrupt every 5.12 ms.
- Timer 3 creates a trigger event every 30 ms.

The PIT module must be activated by writing a 0 to MCR[MDIS].

The 50 MHz clock frequency equates to a clock period of 20 ns. Timer 1 needs to trigger every 5.12 ms/20 ns = 256,000 cycles and Timer 3 every 30 ms/20 ns = 1,500,000 cycles. The value for the LDVAL register trigger is calculated as:

LDVAL trigger = (period / clock period) -1

This means LDVAL1 and LDVAL3 must be written with 0x0003E7FF and 0x0016E35F respectively.

The interrupt for Timer 1 is enabled by setting TCTRL1[TIE]. The timer is started by writing 1 to TCTRL1[TEN].

Timer 3 shall be used only for triggering. Therefore, Timer 3 is started by writing a 1 to TCTRL3[TEN]. TCTRL3[TIE] stays at 0.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
PIT_LDVAL1 = 0x0003E7FF; // setup timer 1 for 256000 cycles
PIT_TCTRL1 = TIE; // enable Timer 1 interrupts
PIT_TCTRL1 |= TEN; // start Timer 1


// Timer 3
PIT_LDVAL3 = 0x0016E35F; // setup timer 3 for 1500000 cycles
PIT_TCTRL3 |= TEN; // start Timer 3
```

40.6 Example configuration for chained timers

In the example configuration:

- The PIT clock has a frequency of 100 MHz.
- Timers 1 and 2 are available.
- An interrupt shall be raised every 1 hour.

The PIT module needs to be activated by writing a 0 to MCR[MDIS].

The 100 MHz clock frequency equates to a clock period of 10 ns, so the PIT needs to count for 6000 million cycles, which is more than a single timer can do. So, Timer 1 is set up to trigger every 6 s (600 million cycles). Timer 2 is chained to Timer 1 and programmed to trigger 10 times.

The value for the LDVAL register trigger is calculated as number of cycles-1, so LDVAL1 receives the value 0x23C345FF and LDVAL2 receives the value 0x00000009.

The interrupt for Timer 2 is enabled by setting TCTRL2[TIE], the Chain mode is activated by setting TCTRL2[CHN], and the timer is started by writing a 1 to TCTRL2[TEN]. TCTRL1[TEN] needs to be set, and TCTRL1[CHN] and TCTRL1[TIE] are cleared.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 2
PIT_LDVAL2 = 0x00000009; // setup Timer 2 for 10 counts
PIT_TCTRL2 = TIE; // enable Timer 2 interrupt
PIT_TCTRL2 |= CHN; // chain Timer 2 to Timer 1
PIT_TCTRL2 |= TEN; // start Timer 2


// Timer 1
PIT_LDVAL1 = 0x23C345FF; // setup Timer 1 for 600 000 000 cycles
PIT_TCTRL1 = TEN; // start Timer 1
```

40.7 Example configuration for the lifetime timer

To configure the lifetimer timer, channels 0 and 1 need to be chained together.

First the PIT module needs to be activated by writing a 0 to the MDIS bit in the CTRL register, then the LDVAL registers need to be set to the maximum value.

The timer is a downcounter.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
PIT_LDVAL1 = 0xFFFFFFFF; // setup timer 1 for maximum counting period
PIT_TCTRL1 = 0x0; // disable timer 1 interrupts
PIT_TCTRL1 |= CHN; // chain timer 1 to timer 0
PIT_TCTRL1 |= TEN; // start timer 1


// Timer 0
PIT_LDVAL0 = 0xFFFFFFFF; // setup timer 0 for maximum counting period
PIT_TCTRL0 = TEN; // start timer 0
```

To access the lifetime, read first LTMR64H and then LTMR64L.

```
current_uptime = PIT_LTMR64H<<32;
current_uptime = current_uptime + PIT_LTMR64L;
```

Chapter 41
Low-Power Timer (LPTMR)

41.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The low-power timer (LPTMR) can be configured to operate as a time counter with optional prescaler, or as a pulse counter with optional glitch filter, across all power modes, including the low-leakage modes. It can also continue operating through most system reset events, allowing it to be used as a time of day counter.

41.1.1 Features

The features of the LPTMR module include:

- 16-bit time counter or pulse counter with compare
  - Optional interrupt can generate asynchronous wakeup from any low-power mode
  - Hardware trigger output
  - Counter supports free-running mode or reset on compare
- Configurable clock source for prescaler/glitch filter
- Configurable input source for pulse counter
  - Rising-edge or falling-edge

41.1.2 Modes of operation

The following table describes the operation of the LPTMR module in various modes.

Table 41-1. Modes of operation

Modes	Description
Run	The LPTMR operates normally.
Wait	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Stop	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Low-Leakage	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Debug	The LPTMR operates normally in Pulse Counter mode, but counter does not increment in Time Counter mode.

41.2 LPTMR signal descriptions

Table 41-2. LPTMR signal descriptions

Signal	I/O	Description
LPTMR_ALT*n*	I	Pulse Counter Input pin

41.2.1 Detailed signal descriptions

Table 41-3. LPTMR interface—detailed signal descriptions

<table>
<tr><th>Signal</th><th>I/O</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">LPTMR_ALTn</td><td rowspan="3">I</td><td colspan="2">Pulse Counter Input
The LPTMR can select one of the input pins to be used in Pulse Counter mode.</td></tr>
<tr><td>State meaning</td><td>Assertion—If configured for pulse counter mode with active-high input, then assertion causes the CNR to increment.
Deassertion—If configured for pulse counter mode with active-low input, then deassertion causes the CNR to increment.</td></tr>
<tr><td>Timing</td><td>Assertion or deassertion may occur at any time; input may assert asynchronously to the bus clock.</td></tr>
</table>

41.3 Memory map and register definition

LPTMR memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_0000	Low Power Timer Control Status Register (LPTMR0_CSR)	32	R/W	0000_0000h	41.3.1/1889
4004_0004	Low Power Timer Prescale Register (LPTMR0_PSR)	32	R/W	0000_0000h	41.3.2/1890
4004_0008	Low Power Timer Compare Register (LPTMR0_CMR)	32	R/W	0000_0000h	41.3.3/1892
4004_000C	Low Power Timer Counter Register (LPTMR0_CNR)	32	R	0000_0000h	41.3.4/1892

41.3.1 Low Power Timer Control Status Register (LPTMRx_CSR)

Address: 4004_0000h base + 0h offset = 4004_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TCF	TIE	TPS		TPP	TFC	TMS	TEN
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CSR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TCF	Timer Compare Flag TCF is set when the LPTMR is enabled and the CNR equals the CMR and increments. TCF is cleared when the LPTMR is disabled or a logic 1 is written to it. 0 The value of CNR is not equal to CMR and increments. 1 The value of CNR is equal to CMR and increments.
6 TIE	Timer Interrupt Enable When TIE is set, the LPTMR Interrupt is generated whenever TCF is also set. 0 Timer interrupt disabled. 1 Timer interrupt enabled.
5–4 TPS	Timer Pin Select Configures the input source to be used in Pulse Counter mode. TPS must be altered only when the LPTMR is disabled. The input connections vary by device. See the chip configuration details for information on the connections to these inputs. 00 Pulse counter input 0 is selected.

Table continues on the next page...

LPTMRx_CSR field descriptions (continued)

Field	Description
	01 Pulse counter input 1 is selected. 10 Pulse counter input 2 is selected. 11 Pulse counter input 3 is selected.
3 TPP	Timer Pin Polarity Configures the polarity of the input source in Pulse Counter mode. TPP must be changed only when the LPTMR is disabled. 0 Pulse Counter input source is active-high, and the CNR will increment on the rising-edge. 1 Pulse Counter input source is active-low, and the CNR will increment on the falling-edge.
2 TFC	Timer Free-Running Counter When clear, TFC configures the CNR to reset whenever TCF is set. When set, TFC configures the CNR to reset on overflow. TFC must be altered only when the LPTMR is disabled. 0 CNR is reset whenever TCF is set. 1 CNR is reset on overflow.
1 TMS	Timer Mode Select Configures the mode of the LPTMR. TMS must be altered only when the LPTMR is disabled. 0 Time Counter mode. 1 Pulse Counter mode.
0 TEN	Timer Enable When TEN is clear, it resets the LPTMR internal logic, including the CNR and TCF. When TEN is set, the LPTMR is enabled. While writing 1 to this field, CSR[5:1] must not be altered. 0 LPTMR is disabled and internal logic is reset. 1 LPTMR is enabled.

41.3.2 Low Power Timer Prescale Register (LPTMRx_PSR)

Address: 4004_0000h base + 4h offset = 4004_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									PRESCALE				PBYP	PCS	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_PSR field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–3 PRESCALE	Prescale Value Configures the size of the Prescaler in Time Counter mode or width of the glitch filter in Pulse Counter mode. PRESCALE must be altered only when the LPTMR is disabled. 0000 Prescaler divides the prescaler clock by 2; glitch filter does not support this configuration. 0001 Prescaler divides the prescaler clock by 4; glitch filter recognizes change on input pin after 2 rising clock edges. 0010 Prescaler divides the prescaler clock by 8; glitch filter recognizes change on input pin after 4 rising clock edges. 0011 Prescaler divides the prescaler clock by 16; glitch filter recognizes change on input pin after 8 rising clock edges. 0100 Prescaler divides the prescaler clock by 32; glitch filter recognizes change on input pin after 16 rising clock edges. 0101 Prescaler divides the prescaler clock by 64; glitch filter recognizes change on input pin after 32 rising clock edges. 0110 Prescaler divides the prescaler clock by 128; glitch filter recognizes change on input pin after 64 rising clock edges. 0111 Prescaler divides the prescaler clock by 256; glitch filter recognizes change on input pin after 128 rising clock edges. 1000 Prescaler divides the prescaler clock by 512; glitch filter recognizes change on input pin after 256 rising clock edges. 1001 Prescaler divides the prescaler clock by 1024; glitch filter recognizes change on input pin after 512 rising clock edges. 1010 Prescaler divides the prescaler clock by 2048; glitch filter recognizes change on input pin after 1024 rising clock edges. 1011 Prescaler divides the prescaler clock by 4096; glitch filter recognizes change on input pin after 2048 rising clock edges. 1100 Prescaler divides the prescaler clock by 8192; glitch filter recognizes change on input pin after 4096 rising clock edges. 1101 Prescaler divides the prescaler clock by 16,384; glitch filter recognizes change on input pin after 8192 rising clock edges. 1110 Prescaler divides the prescaler clock by 32,768; glitch filter recognizes change on input pin after 16,384 rising clock edges. 1111 Prescaler divides the prescaler clock by 65,536; glitch filter recognizes change on input pin after 32,768 rising clock edges.
2 PBYP	Prescaler Bypass When PBYP is set, the selected prescaler clock in Time Counter mode or selected input source in Pulse Counter mode directly clocks the CNR. When PBYP is clear, the CNR is clocked by the output of the prescaler/glitch filter. PBYP must be altered only when the LPTMR is disabled. 0 Prescaler/glitch filter is enabled. 1 Prescaler/glitch filter is bypassed.
1–0 PCS	Prescaler Clock Select Selects the clock to be used by the LPTMR prescaler/glitch filter. PCS must be altered only when the LPTMR is disabled. The clock connections vary by device. **NOTE:** See the chip configuration details for information on the connections to these inputs.

Table continues on the next page...

LPTMRx_PSR field descriptions (continued)

Field	Description
	00 Prescaler/glitch filter clock 0 selected. 01 Prescaler/glitch filter clock 1 selected. 10 Prescaler/glitch filter clock 2 selected. 11 Prescaler/glitch filter clock 3 selected.

41.3.3 Low Power Timer Compare Register (LPTMRx_CMR)

Address: 4004_0000h base + 8h offset = 4004_0008h

Bit	31–16	15–0
R	0	COMPARE
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

LPTMRx_CMR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 COMPARE	Compare Value When the LPTMR is enabled and the CNR equals the value in the CMR and increments, TCF is set and the hardware trigger asserts until the next time the CNR increments. If the CMR is 0, the hardware trigger will remain asserted until the LPTMR is disabled. If the LPTMR is enabled, the CMR must be altered only when TCF is set.

41.3.4 Low Power Timer Counter Register (LPTMRx_CNR)

Address: 4004_0000h base + Ch offset = 4004_000Ch

Bit	31–16	15–0
R	0	COUNTER
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

LPTMRx_CNR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 COUNTER	Counter Value

41.4 Functional description

41.4.1 LPTMR power and reset

The LPTMR remains powered in all power modes, including low-leakage modes. If the LPTMR is not required to remain operating during a low-power mode, then it must be disabled before entering the mode.

The LPTMR is reset only on global Power On Reset (POR) or Low Voltage Detect (LVD). When configuring the LPTMR registers, the CSR must be initially written with the timer disabled, before configuring the PSR and CMR. Then, CSR[TIE] must be set as the last step in the initialization. This ensures the LPTMR is configured correctly and the LPTMR counter is reset to zero following a warm reset.

41.4.2 LPTMR clocking

The LPTMR prescaler/glitch filter can be clocked by one of the four clocks. The clock source must be enabled before the LPTMR is enabled.

NOTE

The clock source selected may need to be configured to remain enabled in low-power modes, otherwise the LPTMR will not operate during low-power modes.

In Pulse Counter mode with the prescaler/glitch filter bypassed, the selected input source directly clocks the CNR and no other clock source is required. To minimize power in this case, configure the prescaler clock source for a clock that is not toggling.

NOTE

The clock source or pulse input source selected for the LPTMR should not exceed the frequency f_{LPTMR} defined in the device datasheet.

41.4.3 LPTMR prescaler/glitch filter

The LPTMR prescaler and glitch filter share the same logic which operates as a prescaler in Time Counter mode and as a glitch filter in Pulse Counter mode.

NOTE

The prescaler/glitch filter configuration must not be altered when the LPTMR is enabled.

41.4.3.1 Prescaler enabled

In Time Counter mode, when the prescaler is enabled, the output of the prescaler directly clocks the CNR. When the LPTMR is enabled, the CNR will increment every 2^2 to 2^{16} prescaler clock cycles. After the LPTMR is enabled, the first increment of the CNR will take an additional one or two prescaler clock cycles due to synchronization logic.

41.4.3.2 Prescaler bypassed

In Time Counter mode, when the prescaler is bypassed, the selected prescaler clock increments the CNR on every clock cycle. When the LPTMR is enabled, the first increment will take an additional one or two prescaler clock cycles due to synchronization logic.

41.4.3.3 Glitch filter

In Pulse Counter mode, when the glitch filter is enabled, the output of the glitch filter directly clocks the CNR. When the LPTMR is first enabled, the output of the glitch filter is asserted, that is, logic 1 for active-high and logic 0 for active-low. The following table shows the change in glitch filter output with the selected input source.

If	Then
The selected input source remains deasserted for at least 2^1 to 2^{15} consecutive prescaler clock rising edges	The glitch filter output will also deassert.
The selected input source remains asserted for at least 2^1 to 2^{15} consecutive prescaler clock rising-edges	The glitch filter output will also assert.

NOTE

The input is only sampled on the rising clock edge.

The CNR will increment each time the glitch filter output asserts. In Pulse Counter mode, the maximum rate at which the CNR can increment is once every 2^2 to 2^{16} prescaler clock edges. When first enabled, the glitch filter will wait an additional one or two prescaler clock edges due to synchronization logic.

41.4.3.4 Glitch filter bypassed

In Pulse Counter mode, when the glitch filter is bypassed, the selected input source increments the CNR every time it asserts. Before the LPTMR is first enabled, the selected input source is forced to be asserted. This prevents the CNR from incrementing if the selected input source is already asserted when the LPTMR is first enabled.

41.4.4 LPTMR compare

When the CNR equals the value of the CMR and increments, the following events occur:

- CSR[TCF] is set.
- LPTMR interrupt is generated if CSR[TIE] is also set.
- LPTMR hardware trigger is generated.
- CNR is reset if CSR[TFC] is clear.

When the LPTMR is enabled, the CMR can be altered only when CSR[TCF] is set. When updating the CMR, the CMR must be written and CSR[TCF] must be cleared before the LPTMR counter has incremented past the new LPTMR compare value.

41.4.5 LPTMR counter

The CNR increments by one on every:

- Prescaler clock in Time Counter mode with prescaler bypassed
- Prescaler output in Time Counter mode with prescaler enabled
- Input source assertion in Pulse Counter mode with glitch filter bypassed
- Glitch filter output in Pulse Counter mode with glitch filter enabled

The CNR is reset when the LPTMR is disabled or if the counter register overflows. If CSR[TFC] is cleared, then the CNR is also reset whenever CSR[TCF] is set.

The CNR continues incrementing when the core is halted in Debug mode when configured for Pulse Counter mode, the CNR will stop incrementing when the core is halted in Debug mode when configured for Time Counter mode.

The CNR cannot be initialized, but can be read at any time. On each read of the CNR, software must first write to the CNR with any value. This will synchronize and register the current value of the CNR into a temporary register. The contents of the temporary register are returned on each read of the CNR.

When reading the CNR, the bus clock must be at least two times faster than the rate at which the LPTMR counter is incrementing, otherwise incorrect data may be returned.

41.4.6 LPTMR hardware trigger

The LPTMR hardware trigger asserts at the same time the CSR[TCF] is set and can be used to trigger hardware events in other peripherals without software intervention. The hardware trigger is always enabled.

When	Then
The CMR is set to 0 with CSR[TFC] clear	The LPTMR hardware trigger will assert on the first compare and does not deassert.
The CMR is set to a nonzero value, or, if CSR[TFC] is set	The LPTMR hardware trigger will assert on each compare and deassert on the following increment of the CNR.

41.4.7 LPTMR interrupt

The LPTMR interrupt is generated whenever CSR[TIE] and CSR[TCF] are set. CSR[TCF] is cleared by disabling the LPTMR or by writing a logic 1 to it.

CSR[TIE] can be altered and CSR[TCF] can be cleared while the LPTMR is enabled.

The LPTMR interrupt is generated asynchronously to the system clock and can be used to generate a wakeup from any low-power mode, including the low-leakage modes, provided the LPTMR is enabled as a wakeup source.

Chapter 42
10/100-Mbps Ethernet MAC (ENET)

42.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The MAC-NET core, in conjunction with a 10/100 MAC, implements layer 3 network acceleration functions. These functions are designed to accelerate the processing of various common networking protocols, such as IP, TCP, UDP, and ICMP, providing wire speed services to client applications.

42.2 Overview

The core implements a dual speed 10/100 Mbit/s Ethernet MAC compliant with the IEEE802.3-2002 standard. The MAC layer provides compatibility with half- or full-duplex 10/100 Mbit/s Ethernet LANs.

The MAC operation is fully programmable and can be used in Network Interface Card (NIC), bridging, or switching applications. The core implements the remote network monitoring (RMON) counters according to IETF RFC 2819.

The core also implements a hardware acceleration block to optimize the performance of network controllers providing TCP/IP, UDP, and ICMP protocol services. The acceleration block performs critical functions in hardware, which are typically implemented with large software overhead.

The core implements programmable embedded FIFOs that can provide buffering on the receive path for lossless flow control.

Advanced power management features are available with magic packet detection and programmable power-down modes.

The programmable 10/100 Ethernet MAC with IEEE 1588 integrates a standard IEEE 802.3 Ethernet MAC with a time-stamping module. The IEEE 1588 standard provides accurate clock synchronization for distributed control nodes for industrial automation applications.

42.2.1 Features

The MAC-NET core includes the following features.

42.2.1.1 Ethernet MAC features

- Implements the full 802.3 specification with preamble/SFD generation, frame padding generation, CRC generation and checking
- Supports zero length Preamble
- Dynamically configurable to support 10/100 Mbit/s operation
- Supports 10/100 Mbit/s full-duplex and configurable half-duplex operation
- Compliant with the AMD magic packet detection with interrupt for node remote power management
- Seamless interface to commercial ethernet PHY devices via one of the following:
  - a 2-bit Reduced MII (RMII) operating at 50 MHz.
- Simple 64-Bit FIFO user-application interface
- CRC-32 checking at full speed with optional forwarding of the frame check sequence (FCS) field to the client
- CRC-32 generation and append on transmit or forwarding of user application provided FCS selectable on a per-frame basis
- In full-duplex mode:
  - Implements automated pause frame (802.3 x31A) generation and termination, providing flow control without user application intervention
  - Pause quanta used to form pause frames — dynamically programmable
  - Pause frame generation additionally controllable by user application offering flexible traffic flow control
  - Optional forwarding of received pause frames to the user application
  - Implements standard flow-control mechanism
- In half-duplex mode: provides full collision support, including jamming, backoff, and automatic retransmission
- Supports VLAN-tagged frames according to IEEE 802.1Q
- Programmable MAC address: Insertion on transmit; discards frames with mismatching destination address on receive (except broadcast and pause frames)
- Programmable promiscuous mode support to omit MAC destination address checking on receive

- Multicast and unicast address filtering on receive based on 64-entry hash table, reducing higher layer processing load
- Programmable frame maximum length providing support for any standard or proprietary frame length
- Statistics indicators for frame traffic and errors (alignment, CRC, length) and pause frames providing for IEEE 802.3 basic and mandatory management information database (MIB) package and remote network monitoring (RFC 2819)
- Simple handshake user application FIFO interface with fully programmable depth and threshold levels
- Provides separate status word for each received frame on the user interface providing information such as frame length, frame type, VLAN tag, and error information
- MII internal loopback
- MDIO master interface for PHY device configuration and management with two programmable MDIO base addresses
- Supports legacy FEC buffer descriptors
- Important Note:The theoretical maximum performance of 1 Gbps ENET is limited to 470 Mbps (total for Tx and Rx) due to Chip internal bus throughput limitation. The actual measured performance in an optimized environment is up to 400 Mbps. See erratum ERR004512 in the device silicon errata document.

42.2.1.2 IP protocol performance optimization features

- Operates on TCP/IP and UDP/IP and ICMP/IP protocol data or IP header only
- Enables wire-speed processing
- Supports IPv4 and IPv6
- Transparent passing of frames of other types and protocols
- Supports VLAN tagged frames according to IEEE 802.1q with transparent forwarding of VLAN tag and control field
- Automatic IP-header and payload (protocol specific) checksum calculation and verification on receive
- Automatic IP-header and payload (protocol specific) checksum generation and automatic insertion on transmit configurable on a per-frame basis
- Supports IP and TCP, UDP, ICMP data for checksum generation and checking
- Supports full header options for IPv4 and TCP protocol headers
- Provides IPv6 support to datagrams with base header only — datagrams with extension headers are passed transparently unmodifed/unchecked

- Provides statistics information for received IP and protocol errors
- Configurable automatic discard of erroneous frames
- Configurable automatic host-to-network (RX) and network-to-host (TX) byte order conversion for IP and TCP/UDP/ICMP headers within the frame
- Configurable padding remove for short IP datagrams on receive
- Configurable Ethernet payload alignment to allow for 32-bit word-aligned header and payload processing
- Programmable store-and-forward operation with clock and rate decoupling FIFOs

42.2.1.3 IEEE 1588 features

- Supports all IEEE 1588 frames
- Allows reference clock to be chosen independently of network speed
- Software-programmable precise time-stamping of ingress and egress frames
- Timer monitoring capabilities for system calibration and timing accuracy management
- Precise time-stamping of external events with programmable interrupt generation
- Programmable event and interrupt generation for external system control
- Supports hardware- and software-controllable timer synchronization
- Provides a 4-channel IEEE 1588 timer — each channel supports input capture and output compare using the 1588 counter

42.2.2 Block diagram

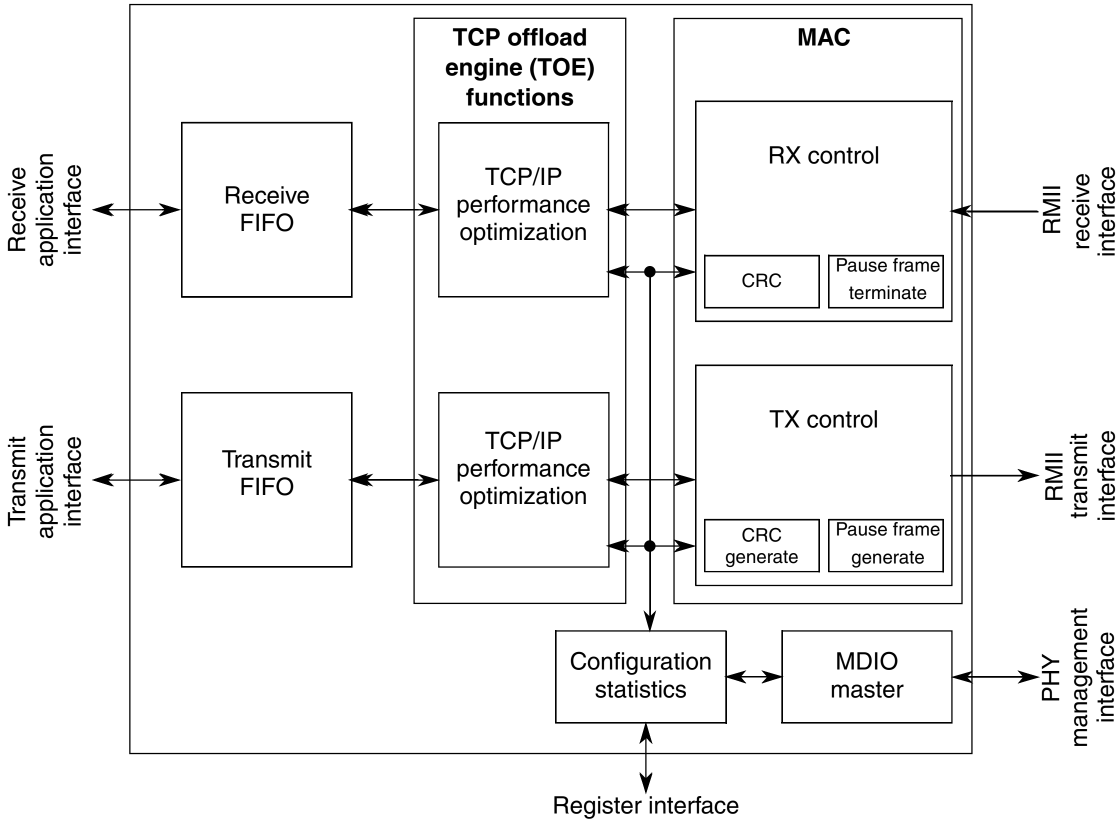


Figure 42-1. 10/100 Ethernet MAC-NET Core Block Diagram

42.3 External signal description

RMII	Description	I/O
—	Asserted upon detection of a collision and remains asserted while the collision persists. This signal is not defined for full-duplex mode.	I
—	Carrier sense. When asserted, indicates transmit or receive medium is not idle. In RMII mode, this signal is present on the RMII_CRS_DV pin.	I
RMII_MDC	Output clock provides a timing reference to the PHY for data transfers on the MDIO signal.	O
RMII_MDIO	Transfers control information between the external PHY and the media-access controller. Data is synchronous to MDC. This signal is an input after reset.	I/O

Table continues on the next page...

RMII	Description	I/O
—	In MII mode, provides a timing reference for RXDV, RXD[3:0], and RXER.	I
RMII_CRS_DV	Asserting this input indicates the PHY has valid nibbles present on the MII. RXDV must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RXDV must start no later than the SFD and exclude any EOF. In RMII mode, this pin also generates the CRS signal.	I
RMII_RXD[1:0]	Contains the Ethernet input data transferred from the PHY to the media-access controller when RXDV is asserted.	I
RMII_RXER	When asserted with RXDV, indicates the PHY detects an error in the current frame.	I
—	Input clock, which provides a timing reference for TXEN, TXD[3:0], and TXER.	I
RMII_TXD[1:0]	Serial output Ethernet data. Only valid during TXEN assertion.	O
RMII_TXEN	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is deasserted before the first TXCLK following the final nibble of the frame.	O
—	When asserted for one or more clock cycles while TXEN is also asserted, PHY sends one or more illegal symbols.	O
RMII_REF_CLK	In RMII mode, this signal is the reference clock for receive, transmit, and the control interface.	I
—	In RGMII mode, this signal provide 125Mhz external reference clock input.	I
1588_TMR*n*	Capture/compare block input/output event bus. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCR*n* register for inspection by software. When configured for compare, the corresponding signal 1588_TMR*n* is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCR*n*. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSR*n*[TIE] or ENET_TCSR*n*[TDRE] is set.	I/O

Table continues on the next page...

RMII	Description	I/O
ENET_1588_CLKIN	Alternate IEEE 1588 Ethernet clock input; Clock period should be an integer number of nanoseconds	I

42.4 Memory map/register definition

Reserved bits should be written with 0 and ignored on read to allow future extension. Unused registers read zero and a write has no effect.

The table found here summarizes the Ethernet registers.

Table 42-1. Register map summary

Offset Address	Section	Description
0x000	Configuration	Core control and status registers
0x200	Statistics counters	MIB block counters. See Statistic event counters.
0x400	1588 control	1588 adjustable timer (TSM) and 1588 frame control
0x600	Capture/compare block	Registers for the capture/compare block

ENET memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_0004	Interrupt Event Register (ENET0_EIR)	32	w1c	0000_0000h	42.4.1/1907
400D_0008	Interrupt Mask Register (ENET0_EIMR)	32	R/W	0000_0000h	42.4.2/1910
400D_0010	Receive Descriptor Active Register (ENET0_RDAR)	32	R/W	0000_0000h	42.4.3/1913
400D_0014	Transmit Descriptor Active Register (ENET0_TDAR)	32	R/W	0000_0000h	42.4.4/1913
400D_0024	Ethernet Control Register (ENET0_ECR)	32	R/W	F000_0000h	42.4.5/1914
400D_0040	MII Management Frame Register (ENET0_MMFR)	32	R/W	0000_0000h	42.4.6/1916
400D_0044	MII Speed Control Register (ENET0_MSCR)	32	R/W	0000_0000h	42.4.7/1917
400D_0064	MIB Control Register (ENET0_MIBC)	32	R/W	C000_0000h	42.4.8/1919
400D_0084	Receive Control Register (ENET0_RCR)	32	R/W	05EE_0001h	42.4.9/1920
400D_00C4	Transmit Control Register (ENET0_TCR)	32	R/W	0000_0000h	42.4.10/ 1923
400D_00E4	Physical Address Lower Register (ENET0_PALR)	32	R/W	0000_0000h	42.4.11/ 1925
400D_00E8	Physical Address Upper Register (ENET0_PAUR)	32	R/W	0000_8808h	42.4.12/ 1925
400D_00EC	Opcode/Pause Duration Register (ENET0_OPD)	32	R/W	0001_0000h	42.4.13/ 1926
400D_0118	Descriptor Individual Upper Address Register (ENET0_IAUR)	32	R/W	0000_0000h	42.4.14/ 1926

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_011C	Descriptor Individual Lower Address Register (ENET0_IALR)	32	R/W	0000_0000h	42.4.15/ 1927
400D_0120	Descriptor Group Upper Address Register (ENET0_GAUR)	32	R/W	0000_0000h	42.4.16/ 1927
400D_0124	Descriptor Group Lower Address Register (ENET0_GALR)	32	R/W	0000_0000h	42.4.17/ 1928
400D_0144	Transmit FIFO Watermark Register (ENET0_TFWR)	32	R/W	0000_0000h	42.4.18/ 1928
400D_0180	Receive Descriptor Ring Start Register (ENET0_RDSR)	32	R/W	0000_0000h	42.4.19/ 1929
400D_0184	Transmit Buffer Descriptor Ring Start Register (ENET0_TDSR)	32	R/W	0000_0000h	42.4.20/ 1930
400D_0188	Maximum Receive Buffer Size Register (ENET0_MRBR)	32	R/W	0000_0000h	42.4.21/ 1930
400D_0190	Receive FIFO Section Full Threshold (ENET0_RSFL)	32	R/W	0000_0000h	42.4.22/ 1931
400D_0194	Receive FIFO Section Empty Threshold (ENET0_RSEM)	32	R/W	0000_0000h	42.4.23/ 1931
400D_0198	Receive FIFO Almost Empty Threshold (ENET0_RAEM)	32	R/W	0000_0004h	42.4.24/ 1932
400D_019C	Receive FIFO Almost Full Threshold (ENET0_RAFL)	32	R/W	0000_0004h	42.4.25/ 1932
400D_01A0	Transmit FIFO Section Empty Threshold (ENET0_TSEM)	32	R/W	0000_0000h	42.4.26/ 1933
400D_01A4	Transmit FIFO Almost Empty Threshold (ENET0_TAEM)	32	R/W	0000_0004h	42.4.27/ 1933
400D_01A8	Transmit FIFO Almost Full Threshold (ENET0_TAFL)	32	R/W	0000_0008h	42.4.28/ 1933
400D_01AC	Transmit Inter-Packet Gap (ENET0_TIPG)	32	R/W	0000_000Ch	42.4.29/ 1934
400D_01B0	Frame Truncation Length (ENET0_FTRL)	32	R/W	0000_07FFh	42.4.30/ 1934
400D_01C0	Transmit Accelerator Function Configuration (ENET0_TACC)	32	R/W	0000_0000h	42.4.31/ 1935
400D_01C4	Receive Accelerator Function Configuration (ENET0_RACC)	32	R/W	0000_0000h	42.4.32/ 1936
400D_0400	Timer Control Register (ENET0_ATCR)	32	R/W	0000_0000h	42.4.33/ 1937
400D_0404	Timer Value Register (ENET0_ATVR)	32	R/W	0000_0000h	42.4.34/ 1939
400D_0408	Timer Offset Register (ENET0_ATOFF)	32	R/W	0000_0000h	42.4.35/ 1939
400D_040C	Timer Period Register (ENET0_ATPER)	32	R/W	3B9A_CA00h	42.4.36/ 1939

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_0410	Timer Correction Register (ENET0_ATCOR)	32	R/W	0000_0000h	42.4.37/ 1940
400D_0414	Time-Stamping Clock Period Register (ENET0_ATINC)	32	R/W	0000_0000h	42.4.38/ 1940
400D_0418	Timestamp of Last Transmitted Frame (ENET0_ATSTMP)	32	R	0000_0000h	42.4.39/ 1941
400D_0604	Timer Global Status Register (ENET0_TGSR)	32	R/W	0000_0000h	42.4.40/ 1941
400D_0608	Timer Control Status Register (ENET0_TCSR0)	32	R/W	0000_0000h	42.4.41/ 1942
400D_060C	Timer Compare Capture Register (ENET0_TCCR0)	32	R/W	0000_0000h	42.4.42/ 1943
400D_0610	Timer Control Status Register (ENET0_TCSR1)	32	R/W	0000_0000h	42.4.41/ 1942
400D_0614	Timer Compare Capture Register (ENET0_TCCR1)	32	R/W	0000_0000h	42.4.42/ 1943
400D_0618	Timer Control Status Register (ENET0_TCSR2)	32	R/W	0000_0000h	42.4.41/ 1942
400D_061C	Timer Compare Capture Register (ENET0_TCCR2)	32	R/W	0000_0000h	42.4.42/ 1943
400D_0620	Timer Control Status Register (ENET0_TCSR3)	32	R/W	0000_0000h	42.4.41/ 1942
400D_0624	Timer Compare Capture Register (ENET0_TCCR3)	32	R/W	0000_0000h	42.4.42/ 1943
400D_1004	Interrupt Event Register (ENET1_EIR)	32	w1c	0000_0000h	42.4.1/1907
400D_1008	Interrupt Mask Register (ENET1_EIMR)	32	R/W	0000_0000h	42.4.2/1910
400D_1010	Receive Descriptor Active Register (ENET1_RDAR)	32	R/W	0000_0000h	42.4.3/1913
400D_1014	Transmit Descriptor Active Register (ENET1_TDAR)	32	R/W	0000_0000h	42.4.4/1913
400D_1024	Ethernet Control Register (ENET1_ECR)	32	R/W	F000_0000h	42.4.5/1914
400D_1040	MII Management Frame Register (ENET1_MMFR)	32	R/W	0000_0000h	42.4.6/1916
400D_1044	MII Speed Control Register (ENET1_MSCR)	32	R/W	0000_0000h	42.4.7/1917
400D_1064	MIB Control Register (ENET1_MIBC)	32	R/W	C000_0000h	42.4.8/1919
400D_1084	Receive Control Register (ENET1_RCR)	32	R/W	05EE_0001h	42.4.9/1920
400D_10C4	Transmit Control Register (ENET1_TCR)	32	R/W	0000_0000h	42.4.10/ 1923
400D_10E4	Physical Address Lower Register (ENET1_PALR)	32	R/W	0000_0000h	42.4.11/ 1925
400D_10E8	Physical Address Upper Register (ENET1_PAUR)	32	R/W	0000_8808h	42.4.12/ 1925
400D_10EC	Opcode/Pause Duration Register (ENET1_OPD)	32	R/W	0001_0000h	42.4.13/ 1926
400D_1118	Descriptor Individual Upper Address Register (ENET1_IAUR)	32	R/W	0000_0000h	42.4.14/ 1926

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_111C	Descriptor Individual Lower Address Register (ENET1_IALR)	32	R/W	0000_0000h	42.4.15/ 1927
400D_1120	Descriptor Group Upper Address Register (ENET1_GAUR)	32	R/W	0000_0000h	42.4.16/ 1927
400D_1124	Descriptor Group Lower Address Register (ENET1_GALR)	32	R/W	0000_0000h	42.4.17/ 1928
400D_1144	Transmit FIFO Watermark Register (ENET1_TFWR)	32	R/W	0000_0000h	42.4.18/ 1928
400D_1180	Receive Descriptor Ring Start Register (ENET1_RDSR)	32	R/W	0000_0000h	42.4.19/ 1929
400D_1184	Transmit Buffer Descriptor Ring Start Register (ENET1_TDSR)	32	R/W	0000_0000h	42.4.20/ 1930
400D_1188	Maximum Receive Buffer Size Register (ENET1_MRBR)	32	R/W	0000_0000h	42.4.21/ 1930
400D_1190	Receive FIFO Section Full Threshold (ENET1_RSFL)	32	R/W	0000_0000h	42.4.22/ 1931
400D_1194	Receive FIFO Section Empty Threshold (ENET1_RSEM)	32	R/W	0000_0000h	42.4.23/ 1931
400D_1198	Receive FIFO Almost Empty Threshold (ENET1_RAEM)	32	R/W	0000_0004h	42.4.24/ 1932
400D_119C	Receive FIFO Almost Full Threshold (ENET1_RAFL)	32	R/W	0000_0004h	42.4.25/ 1932
400D_11A0	Transmit FIFO Section Empty Threshold (ENET1_TSEM)	32	R/W	0000_0000h	42.4.26/ 1933
400D_11A4	Transmit FIFO Almost Empty Threshold (ENET1_TAEM)	32	R/W	0000_0004h	42.4.27/ 1933
400D_11A8	Transmit FIFO Almost Full Threshold (ENET1_TAFL)	32	R/W	0000_0008h	42.4.28/ 1933
400D_11AC	Transmit Inter-Packet Gap (ENET1_TIPG)	32	R/W	0000_000Ch	42.4.29/ 1934
400D_11B0	Frame Truncation Length (ENET1_FTRL)	32	R/W	0000_07FFh	42.4.30/ 1934
400D_11C0	Transmit Accelerator Function Configuration (ENET1_TACC)	32	R/W	0000_0000h	42.4.31/ 1935
400D_11C4	Receive Accelerator Function Configuration (ENET1_RACC)	32	R/W	0000_0000h	42.4.32/ 1936
400D_1400	Timer Control Register (ENET1_ATCR)	32	R/W	0000_0000h	42.4.33/ 1937
400D_1404	Timer Value Register (ENET1_ATVR)	32	R/W	0000_0000h	42.4.34/ 1939
400D_1408	Timer Offset Register (ENET1_ATOFF)	32	R/W	0000_0000h	42.4.35/ 1939
400D_140C	Timer Period Register (ENET1_ATPER)	32	R/W	3B9A_CA00h	42.4.36/ 1939

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_1410	Timer Correction Register (ENET1_ATCOR)	32	R/W	0000_0000h	42.4.37/ 1940
400D_1414	Time-Stamping Clock Period Register (ENET1_ATINC)	32	R/W	0000_0000h	42.4.38/ 1940
400D_1418	Timestamp of Last Transmitted Frame (ENET1_ATSTMP)	32	R	0000_0000h	42.4.39/ 1941
400D_1604	Timer Global Status Register (ENET1_TGSR)	32	R/W	0000_0000h	42.4.40/ 1941
400D_1608	Timer Control Status Register (ENET1_TCSR0)	32	R/W	0000_0000h	42.4.41/ 1942
400D_160C	Timer Compare Capture Register (ENET1_TCCR0)	32	R/W	0000_0000h	42.4.42/ 1943
400D_1610	Timer Control Status Register (ENET1_TCSR1)	32	R/W	0000_0000h	42.4.41/ 1942
400D_1614	Timer Compare Capture Register (ENET1_TCCR1)	32	R/W	0000_0000h	42.4.42/ 1943
400D_1618	Timer Control Status Register (ENET1_TCSR2)	32	R/W	0000_0000h	42.4.41/ 1942
400D_161C	Timer Compare Capture Register (ENET1_TCCR2)	32	R/W	0000_0000h	42.4.42/ 1943
400D_1620	Timer Control Status Register (ENET1_TCSR3)	32	R/W	0000_0000h	42.4.41/ 1942
400D_1624	Timer Compare Capture Register (ENET1_TCCR3)	32	R/W	0000_0000h	42.4.42/ 1943

42.4.1 Interrupt Event Register (ENETx_EIR)

When an event occurs that sets a bit in EIR, an interrupt occurs if the corresponding bit in the interrupt mask register (EIMR) is also set. Writing a 1 to an EIR bit clears it; writing 0 has no effect. This register is cleared upon hardware reset.

NOTE

TxBD[INT] and RxBD[INT] must be set to 1 to allow setting the corresponding EIR register flags in enhanced mode, ENET_ECR[EN1588] = 1. Legacy mode does not require these flags to be enabled.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	PLR	WAKEUP	TS_AVAIL
W		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TS_TIMER															
W	w1c	0		0	0											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_EIR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 BABR	Babbling Receive Error Indicates a frame was received with length in excess of RCR[MAX_FL] bytes.
29 BABT	Babbling Transmit Error Indicates the transmitted frame length exceeds RCR[MAX_FL] bytes. Usually this condition is caused when a frame that is too long is placed into the transmit data buffer(s). Truncation does not occur.
28 GRA	Graceful Stop Complete This interrupt is asserted after the transmitter is put into a pause state after completion of the frame currently being transmitted. See Graceful Transmit Stop (GTS) for conditions that lead to graceful stop. **NOTE:** The GRA interrupt is asserted only when the TX transitions into the stopped state. If this bit is cleared by writing 1 and the TX is still stopped, the bit is not set again.
27 TXF	Transmit Frame Interrupt Indicates a frame has been transmitted and the last corresponding buffer descriptor has been updated.
26 TXB	Transmit Buffer Interrupt Indicates a transmit buffer descriptor has been updated.
25 RXF	Receive Frame Interrupt Indicates a frame has been received and the last corresponding buffer descriptor has been updated.

Table continues on the next page...

ENETx_EIR field descriptions (continued)

Field	Description
24 RXB	Receive Buffer Interrupt Indicates a receive buffer descriptor is not the last in the frame has been updated.
23 MII	MII Interrupt. Indicates that the MII has completed the data transfer requested.
22 EBERR	Ethernet Bus Error Indicates a system bus error occurred when a uDMA transaction is underway. When this bit is set, ECR[ETHEREN] is cleared, halting frame processing by the MAC. When this occurs, software must ensure proper actions, possibly resetting the system, to resume normal operation.
21 LC	Late Collision Indicates a collision occurred beyond the collision window (slot time) in half-duplex mode. The frame truncates with a bad CRC and the remainder of the frame is discarded.
20 RL	Collision Retry Limit Indicates a collision occurred on each of 16 successive attempts to transmit the frame. The frame is discarded without being transmitted and transmission of the next frame commences. This error can only occur in half-duplex mode.
19 UN	Transmit FIFO Underrun Indicates the transmit FIFO became empty before the complete frame was transmitted. A bad CRC is appended to the frame fragment and the remainder of the frame is discarded.
18 PLR	Payload Receive Error Indicates a frame was received with a payload length error. See Frame Length/Type Verification: Payload Length Check for more information.
17 WAKEUP	Node Wakeup Request Indication Read-only status bit to indicate that a magic packet has been detected. Will act only if ECR[MAGICEN] is set.
16 TS_AVAIL	Transmit Timestamp Available Indicates that the timestamp of the last transmitted timing frame is available in the ATSTMP register.
15 TS_TIMER	Timestamp Timer The adjustable timer reached the period event. A period event interrupt can be generated if ATCR[PEREN] is set and the timer wraps according to the periodic setting in the ATPER register. Set the timer period value before setting ATCR[PEREN].
14–13 Reserved	This field is reserved.
12 Reserved	This field is reserved.
11–0 Reserved	This field is reserved.

42.4.2 Interrupt Mask Register (ENETx_EIMR)

EIMR controls which interrupt events are allowed to generate actual interrupts. A hardware reset clears this register. If the corresponding bits in the EIR and EIMR registers are set, an interrupt is generated. The interrupt signal remains asserted until a 1 is written to the EIR field (write 1 to clear) or a 0 is written to the EIMR field.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	PLR	WAKEUP	TS_AVAIL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TS_TIMER	0		0												
W					0											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_EIMR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 BABR	BABR Interrupt Mask Corresponds to interrupt source EIR[BABR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
29 BABT	BABT Interrupt Mask Corresponds to interrupt source EIR[BABT] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABT field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
28 GRA	GRA Interrupt Mask Corresponds to interrupt source EIR[GRA] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The

Table continues on the next page...

ENETx_EIMR field descriptions (continued)

Field	Description
	corresponding EIR GRA field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
27 TXF	TXF Interrupt Mask Corresponds to interrupt source EIR[TXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
26 TXB	TXB Interrupt Mask Corresponds to interrupt source EIR[TXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
25 RXF	RXF Interrupt Mask Corresponds to interrupt source EIR[RXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
24 RXB	RXB Interrupt Mask Corresponds to interrupt source EIR[RXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXB field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
23 MII	MII Interrupt Mask Corresponds to interrupt source EIR[MII] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR MII field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
22 EBERR	EBERR Interrupt Mask Corresponds to interrupt source EIR[EBERR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR EBERR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
21 LC	LC Interrupt Mask Corresponds to interrupt source EIR[LC] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR LC field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

Table continues on the next page...

ENETx_EIMR field descriptions (continued)

Field	Description
20 RL	RL Interrupt Mask Corresponds to interrupt source EIR[RL] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
19 UN	UN Interrupt Mask Corresponds to interrupt source EIR[UN] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR UN field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
18 PLR	PLR Interrupt Mask Corresponds to interrupt source EIR[PLR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR PLR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
17 WAKEUP	WAKEUP Interrupt Mask Corresponds to interrupt source EIR[WAKEUP] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR WAKEUP field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
16 TS_AVAIL	TS_AVAIL Interrupt Mask Corresponds to interrupt source EIR[TS_AVAIL] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_AVAIL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
15 TS_TIMER	TS_TIMER Interrupt Mask Corresponds to interrupt source EIR[TS_TIMER] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_TIMER field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
14–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 Reserved	This field is reserved.

42.4.3 Receive Descriptor Active Register (ENETx_RDAR)

RDAR is a command register, written by the user, to indicate that the receive descriptor ring has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							RDAR	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_RDAR field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

42.4.4 Transmit Descriptor Active Register (ENETx_TDAR)

The TDAR is a command register that the user writes to indicate that the transmit descriptor ring has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							TDAR	0							
W								TDAR								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_TDAR field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
23–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

42.4.5 Ethernet Control Register (ENETx_ECR)

ECR is a read/write user register, though hardware may also alter fields in this register. It controls many of the high level features of the Ethernet MAC, including legacy FEC support through the EN1588 field.

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0	0	0	DBSWP	Reserved	DBGEN	0	EN1588	SLEEP	MAGICEN	ETHEREN	RESET
W								DBSWP	Reserved	DBGEN		EN1588	SLEEP	MAGICEN	ETHEREN	RESET
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_ECR field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 DBSWP	Descriptor Byte Swapping Enable Swaps the byte locations of the buffer descriptors. **NOTE:** This field must be written to 1 after reset. 0 The buffer descriptor bytes are not swapped to support big-endian devices. 1 The buffer descriptor bytes are swapped to support little-endian devices.
7 Reserved	This field is reserved. This field must be set to zero.
6 DBGEN	Debug Enable Enables the MAC to enter hardware freeze mode when the device enters debug mode. 0 MAC continues operation in debug mode. 1 MAC enters hardware freeze mode when the processor is in debug mode.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 EN1588	EN1588 Enable Enables enhanced functionality of the MAC. 0 Legacy FEC buffer descriptors and functions enabled. 1 Enhanced frame time-stamping functions enabled.
3 SLEEP	Sleep Mode Enable 0 Normal operating mode. 1 Sleep mode.
2 MAGICEN	Magic Packet Detection Enable Enables/disables magic packet detection. **NOTE:** MAGICEN is relevant only if the SLEEP field is set. If MAGICEN is set, changing the SLEEP field enables/disables sleep mode and magic packet detection. 0 Magic detection logic disabled. 1 The MAC core detects magic packets and asserts EIR[WAKEUP] when a frame is detected.
1 ETHEREN	Ethernet Enable Enables/disables the Ethernet MAC. When the MAC is disabled, the buffer descriptors for an aborted transmit frame are not updated. The uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers. Hardware clears this field under the following conditions:

Table continues on the next page...

ENETx_ECR field descriptions (continued)

Field	Description
	• RESET is set by software • An error condition causes the EBERR field to set. **NOTE:** ETHEREN must be set at the very last step during ENET configuration/setup/initialization, only *after* all other ENET-related registers have been configured. 0 Reception immediately stops and transmission stops after a bad CRC is appended to any currently transmitted frame. 1 MAC is enabled, and reception and transmission are possible.
0 RESET	Ethernet MAC Reset When this field is set, it clears the ETHEREN field.

42.4.6 MII Management Frame Register (ENETx_MMFR)

Writing to MMFR triggers a management frame transaction to the PHY device unless MSCR is programmed to zero.

If MSCR is changed from zero to non-zero during a write to MMFR, an MII frame is generated with the data previously written to the MMFR. This allows MMFR and MSCR to be programmed in either order if MSCR is currently zero.

If the MMFR register is written while frame generation is in progress, the frame contents are altered. Software must use the EIR[MII] interrupt indication to avoid writing to the MMFR register while frame generation is in progress.

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ST		OP		PA					RA					TA		DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_MMFR field descriptions

Field	Description
31–30 ST	Start Of Frame Delimiter These fields must be programmed to 01 for a valid MII management frame.
29–28 OP	Operation Code Determines the frame operation. 00 Write frame operation, but not MII compliant. 01 Write frame operation for a valid MII management frame. 10 Read frame operation for a valid MII management frame. 11 Read frame operation, but not MII compliant.

Table continues on the next page...

ENETx_MMFR field descriptions (continued)

Field	Description
27–23 PA	PHY Address Specifies one of up to 32 attached PHY devices.
22–18 RA	Register Address Specifies one of up to 32 registers within the specified PHY device.
17–16 TA	Turn Around This field must be programmed to 10 to generate a valid MII management frame.
15–0 DATA	Management Frame Data This is the field for data to be written to or read from the PHY register.

42.4.7 MII Speed Control Register (ENETx_MSCR)

MSCR provides control of the MII clock (MDC pin) frequency and allows a preamble drop on the MII management frame.

The MII_SPEED field must be programmed with a value to provide an MDC frequency of less than or equal to 2.5 MHz to be compliant with the IEEE 802.3 MII specification. The MII_SPEED must be set to a non-zero value to source a read or write management frame. After the management frame is complete, the MSCR register may optionally be cleared to turn off MDC. The MDC signal generated has a 50% duty cycle except when MII_SPEED changes during operation. This change takes effect following a rising or falling edge of MDC.

If the internal module clock is 25 MHz, programming this register to 0x0000_0004 results in an MDC as stated in the following equation:

25 MHz / ((4 + 1) x 2) = 2.5 MHz

The following table shows the optimum values for MII_SPEED as a function of internal module clock frequency.

Table 42-62. Programming Examples for MSCR

Internal MAC clock frequency	MSCR [MII_SPEED]	MDC frequency
25 MHz	0x4	2.50 MHz
33 MHz	0x6	2.36 MHz
40 MHz	0x7	2.50 MHz
50 MHz	0x9	2.50 MHz
66 MHz	0xD	2.36 MHz

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					HOLDTIME			DIS_PRE	MII_SPEED						0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_MSCR field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 HOLDTIME	Holdtime On MDIO Output IEEE802.3 clause 22 defines a minimum of 10 ns for the holdtime on the MDIO output. Depending on the host bus frequency, the setting may need to be increased. 000 1 internal module clock cycle 001 2 internal module clock cycles 010 3 internal module clock cycles 111 8 internal module clock cycles
7 DIS_PRE	Disable Preamble Enables/disables prepending a preamble to the MII management frame. The MII standard allows the preamble to be dropped if the attached PHY devices do not require it. 0 Preamble enabled. 1 Preamble (32 ones) is not prepended to the MII management frame.
6–1 MII_SPEED	MII Speed Controls the frequency of the MII management interface clock (MDC) relative to the internal module clock. A value of 0 in this field turns off MDC and leaves it in low voltage state. Any non-zero value results in the MDC frequency of: 1/((MII_SPEED + 1) x 2) of the internal module clock frequency
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

42.4.8 MIB Control Register (ENETx_MIBC)

MIBC is a read/write register controlling and observing the state of the MIB block. Access this register to disable the MIB block operation or clear the MIB counters. The MIB_DIS field resets to 1.

Address: Base address + 64h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MIB_DIS	MIB_IDLE	MIB_CLEAR	0												
W																
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_MIBC field descriptions

Field	Description
31 MIB_DIS	Disable MIB Logic If this control field is set, the MIB logic halts and does not update any MIB counters.
30 MIB_IDLE	MIB Idle If this status field is set, the MIB block is not currently updating any MIB counters.
29 MIB_CLEAR	MIB Clear

Table continues on the next page...

ENETx_MIBC field descriptions (continued)

Field	Description
	If set, all statistics counters are reset to 0. **NOTE:** This field is not self-clearing. To clear the MIB counters set and then clear the field.
28–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

42.4.9 Receive Control Register (ENETx_RCR)

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GRS	NLC	MAX_FL													
W		NLC	MAX_FL													
Reset	0	0	0	0	0	1	0	1	1	1	1	0	1	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CFEN	CRCFWD	PAUFWD	PADEN	0		RMII_10T	RMII_MODE	0	0	FCE	BC_REJ	PROM	MII_MODE	DRT	LOOP
W	CFEN	CRCFWD	PAUFWD	PADEN			RMII_10T	RMII_MODE			FCE	BC_REJ	PROM	MII_MODE	DRT	LOOP
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

ENETx_RCR field descriptions

Field	Description
31 GRS	Graceful Receive Stopped Read-only status indicating that the MAC receive datapath is stopped.

Table continues on the next page...

ENETx_RCR field descriptions (continued)

Field	Description
30 NLC	Payload Length Check Disable Enables/disables a payload length check. 0 The payload length check is disabled. 1 The core checks the frame's payload length with the frame length/type field. Errors are indicated in the EIR[PLC] field.
29–16 MAX_FL	Maximum Frame Length Resets to decimal 1518. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL cause the BABT interrupt to occur. Receive frames longer than MAX_FL cause the BABR interrupt to occur and set the LG field in the end of frame receive buffer descriptor. The recommended default value to be programmed is 1518 or 1522 if VLAN tags are supported.
15 CFEN	MAC Control Frame Enable Enables/disables the MAC control frame. 0 MAC control frames with any opcode other than 0x0001 (pause frame) are accepted and forwarded to the client interface. 1 MAC control frames with any opcode other than 0x0001 (pause frame) are silently discarded.
14 CRCFWD	Terminate/Forward Received CRC Specifies whether the CRC field of received frames is transmitted or stripped. **NOTE:** If padding function is enabled (PADEN = 1), CRCFWD is ignored and the CRC field is checked and always terminated and removed. 0 The CRC field of received frames is transmitted to the user application. 1 The CRC field is stripped from the frame.
13 PAUFWD	Terminate/Forward Pause Frames Specifies whether pause frames are terminated or forwarded. 0 Pause frames are terminated and discarded in the MAC. 1 Pause frames are forwarded to the user application.
12 PADEN	Enable Frame Padding Remove On Receive Specifies whether the MAC removes padding from received frames. 0 No padding is removed on receive by the MAC. 1 Padding is removed from received frames.
11–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 RMII_10T	Enables 10-Mbps mode of the RMII . 0 100 Mbps operation. 1 10 Mbps operation.
8 RMII_MODE	RMII Mode Enable Specifies whether the MAC is configured for MII mode or RMII operation .

Table continues on the next page...

ENETx_RCR field descriptions (continued)

Field	Description
	0 MAC configured for MII mode. 1 MAC configured for RMII operation.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 FCE	Flow Control Enable If set, the receiver detects PAUSE frames. Upon PAUSE frame detection, the transmitter stops transmitting data frames for a given duration.
4 BC_REJ	Broadcast Frame Reject If set, frames with destination address (DA) equal to 0xFFFF_FFFF_FFFF are rejected unless the PROM field is set. If BC_REJ and PROM are set, frames with broadcast DA are accepted and the MISS (M) is set in the receive buffer descriptor.
3 PROM	Promiscuous Mode All frames are accepted regardless of address matching. 0 Disabled. 1 Enabled.
2 MII_MODE	Media Independent Interface Mode This field must always be set. 0 Reserved. 1 MII or RMII mode, as indicated by the RMII_MODE field.
1 DRT	Disable Receive On Transmit 0 Receive path operates independently of transmit. Used for full-duplex or to monitor transmit activity in half-duplex mode. 1 Disable reception of frames while transmitting. Normally used for half-duplex mode.
0 LOOP	Internal Loopback This is an MII internal loopback, therefore MII_MODE must be written to 1 and RMII_MODE must be written to 0. 0 Loopback disabled. 1 Transmitted frames are looped back internal to the device and transmit MII output signals are not asserted. DRT must be cleared.

42.4.10 Transmit Control Register (ENETx_TCR)

TCR is read/write and configures the transmit block. This register is cleared at system reset. FDEN can only be modified when ECR[ETHEREN] is cleared.

Address: Base address + C4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						CRCFWD	ADDINS	ADDSEL			RFC_PAUSE	TFC_PAUSE	FDEN	0	GTS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_TCR field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 CRCFWD	Forward Frame From Application With CRC 0 TxBD[TC] controls whether the frame has a CRC from the application. 1 The transmitter does not append any CRC to transmitted frames, as it is expecting a frame with CRC from the application.
8 ADDINS	Set MAC Address On Transmit

Table continues on the next page...

ENETx_TCR field descriptions (continued)

Field	Description
	0 The source MAC address is not modified by the MAC. 1 The MAC overwrites the source MAC address with the programmed MAC address according to ADDSEL.
7–5 ADDSEL	Source MAC Address Select On Transmit If ADDINS is set, indicates the MAC address that overwrites the source MAC address. 000 Node MAC address programmed on PADDR1/2 registers. 100 Reserved. 101 Reserved. 110 Reserved.
4 RFC_PAUSE	Receive Frame Control Pause This status field is set when a full-duplex flow control pause frame is received and the transmitter pauses for the duration defined in this pause frame. This field automatically clears when the pause duration is complete.
3 TFC_PAUSE	Transmit Frame Control Pause Pauses frame transmission. When this field is set, EIR[GRA] is set. With transmission of data frames stopped, the MAC transmits a MAC control PAUSE frame. Next, the MAC clears TFC_PAUSE and resumes transmitting data frames. If the transmitter pauses due to user assertion of GTS or reception of a PAUSE frame, the MAC may continue transmitting a MAC control PAUSE frame. 0 No PAUSE frame transmitted. 1 The MAC stops transmission of data frames after the current transmission is complete.
2 FDEN	Full-Duplex Enable If this field is set, frames transmit independent of carrier sense and collision inputs. Only modify this bit when ECR[ETHEREN] is cleared.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 GTS	Graceful Transmit Stop When this field is set, MAC stops transmission after any frame currently transmitted is complete and EIR[GRA] is set. If frame transmission is not currently underway, the GRA interrupt is asserted immediately. After transmission finishes, clear GTS to restart. The next frame in the transmit FIFO is then transmitted. If an early collision occurs during transmission when GTS is set, transmission stops after the collision. The frame is transmitted again after GTS is cleared. There may be old frames in the transmit FIFO that transmit when GTS is reasserted. To avoid this, clear ECR[ETHEREN] following the GRA interrupt.

42.4.11 Physical Address Lower Register (ENETx_PALR)

PALR contains the lower 32 bits (bytes 0, 1, 2, 3) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 0 through 3 of the six-byte source address field when transmitting PAUSE frames. This register is not reset and you must initialize it.

Address: Base address + E4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PADDR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_PALR field descriptions

Field	Description
31–0 PADDR1	Pause Address Bytes 0 (bits 31:24), 1 (bits 23:16), 2 (bits 15:8), and 3 (bits 7:0) of the 6-byte individual address are used for exact match and the source address field in PAUSE frames.

42.4.12 Physical Address Upper Register (ENETx_PAUR)

PAUR contains the upper 16 bits (bytes 4 and 5) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 4 and 5 of the six-byte source address field when transmitting PAUSE frames. Bits 15:0 of PAUR contain a constant type field (0x8808) for transmission of PAUSE frames. The upper 16 bits of this register are not reset and you must initialize it.

Address: Base address + E8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PADDR2																TYPE															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

ENETx_PAUR field descriptions

Field	Description
31–16 PADDR2	Bytes 4 (bits 31:24) and 5 (bits 23:16) of the 6-byte individual address used for exact match, and the source address field in PAUSE frames.
15–0 TYPE	Type Field In PAUSE Frames These fields have a constant value of 0x8808.

42.4.13 Opcode/Pause Duration Register (ENETx_OPD)

OPD is read/write accessible. This register contains the 16-bit opcode and 16-bit pause duration fields used in transmission of a PAUSE frame. The opcode field is a constant value, 0x0001. When another node detects a PAUSE frame, that node pauses transmission for the duration specified in the pause duration field. The lower 16 bits of this register are not reset and you must initialize it.

Address: Base address + ECh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OPCODE																PAUSE_DUR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_OPD field descriptions

Field	Description
31–16 OPCODE	Opcode Field In PAUSE Frames These fields have a constant value of 0x0001.
15–0 PAUSE_DUR	Pause Duration Pause duration field used in PAUSE frames.

42.4.14 Descriptor Individual Upper Address Register (ENETx_IAUR)

IAUR contains the upper 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the destination address (DA) field of receive frames with an individual DA. This register is not reset and you must initialize it.

Address: Base address + 118h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_IAUR field descriptions

Field	Description
31–0 IADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR1 contains hash index bit 63. Bit 0 of IADDR1 contains hash index bit 32.

42.4.15 Descriptor Individual Lower Address Register (ENETx_IALR)

IALR contains the lower 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the DA field of receive frames with an individual DA. This register is not reset and you must initialize it.

Address: Base address + 11Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_IALR field descriptions

Field	Description
31–0 IADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR2 contains hash index bit 31. Bit 0 of IADDR2 contains hash index bit 0.

42.4.16 Descriptor Group Upper Address Register (ENETx_GAUR)

GAUR contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

Address: Base address + 120h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_GAUR field descriptions

Field	Description
31–0 GADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR1 contains hash index bit 63. Bit 0 of GADDR1 contains hash index bit 32.

42.4.17 Descriptor Group Lower Address Register (ENETx_GALR)

GALR contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

Address: Base address + 124h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_GALR field descriptions

Field	Description
31–0 GADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR2 contains hash index bit 31. Bit 0 of GADDR2 contains hash index bit 0.

42.4.18 Transmit FIFO Watermark Register (ENETx_TFWR)

If TFR[STRFWD] is cleared, TFWR[TFWR] controls the amount of data required in the transmit FIFO before transmission of a frame can begin. This allows you to minimize transmit latency (TFWR = 00 or 01) or allow for larger bus access latency (TFWR = 11) due to contention for the system bus. Setting the watermark to a high value minimizes the risk of transmit FIFO underrun due to contention for the system bus. The byte counts associated with the TFWR field may need to be modified to match a given system requirement. For example, worst case bus access latency by the transmit data DMA channel.

Address: Base address + 144h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							STRFWD	0		TFWR					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_TFWR field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 STRFWD	Store And Forward Enable 0 Disabled, the transmission start threshold is programmed in TFWR. 1 Enabled.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 TFWR	Transmit FIFO Write If STRFWD is cleared, this field indicates the number of bytes written to the transmit FIFO before transmission of a frame begins. **NOTE:** If a frame with less than the threshold is written, it is still sent independently of this threshold setting. The threshold is relevant only if the frame is larger than the threshold given. This chip may not support the maximum number of bytes written shown below. See the chip-specific information for the ENET module for this value. 000000 64 bytes written. 000001 64 bytes written. 000010 128 bytes written. 000011 192 bytes written. ... ... 111110 3968 bytes written. 111111 4032 bytes written.

42.4.19 Receive Descriptor Ring Start Register (ENETx_RDSR)

RDSR points to the beginning of the circular receive buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, it is recommended to be 128-bit aligned, that is, evenly divisible by 16.

NOTE

This register must be initialized prior to operation

Address: Base address + 180h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R_DES_START																													0		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_RDSR field descriptions

Field	Description
31–3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue.

Table continues on the next page...

ENETx_RDSR field descriptions (continued)

Field	Description
2–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

42.4.20 Transmit Buffer Descriptor Ring Start Register (ENETx_TDSR)

TDSR provides a pointer to the beginning of the circular transmit buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, it is recommended to be 128-bit aligned, that is, evenly divisible by 16.

NOTE

This register must be initialized prior to operation.

Address: Base address + 184h offset

Bit	31–3	2–0
R	X_DES_START	0
W	X_DES_START	

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_TDSR field descriptions

Field	Description
31–3 X_DES_START	Pointer to the beginning of the transmit buffer descriptor queue.
2–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

42.4.21 Maximum Receive Buffer Size Register (ENETx_MRBR)

The MRBR is a user-programmable register that dictates the maximum size of all receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- To allow one maximum size frame per buffer, MRBR must be set to RCR[MAX_FL] or larger.
- To properly align the buffer, MRBR must be evenly divisible by 16. To ensure this, bits 3–0 are set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR greater than or equal to 256 bytes.

NOTE

This register must be initialized before operation.

Address: Base address + 188h offset

Bit	31–14	13–4	3–0
R	0	R_BUF_SIZE	0
W			
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0

ENETx_MRBR field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–4 R_BUF_SIZE	Receive buffer size in bytes.
3–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

42.4.22 Receive FIFO Section Full Threshold (ENETx_RSFL)

Address: Base address + 190h offset

Bit	31–8	7–0
R	0	RX_SECTION_FULL
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

ENETx_RSFL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 RX_SECTION_FULL	Value Of Receive FIFO Section Full Threshold Value, in 64-bit words, of the receive FIFO section full threshold. Clear this field to enable store and forward on the RX FIFO. When programming a value greater than 0 (cut-through operation), it must be greater than RAEM[RX_ALMOST_EMPTY].

42.4.23 Receive FIFO Section Empty Threshold (ENETx_RSEM)

Address: Base address + 194h offset

Bit	31–21	20–16	15–8	7–0
R	0	0	0	RX_SECTION_EMPTY
W				
Reset	0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

ENETx_RSEM field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 RX_SECTION_ EMPTY	Value Of The Receive FIFO Section Empty Threshold Value, in 64-bit words, of the receive FIFO section empty threshold.

42.4.24 Receive FIFO Almost Empty Threshold (ENETx_RAEM)

Address: Base address + 198h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								RX_ALMOST_EMPTY							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

ENETx_RAEM field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 RX_ALMOST_ EMPTY	Value Of The Receive FIFO Almost Empty Threshold Value, in 64-bit words, of the receive FIFO almost empty threshold.

42.4.25 Receive FIFO Almost Full Threshold (ENETx_RAFL)

Address: Base address + 19Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								RX_ALMOST_FULL							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

ENETx_RAFL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 RX_ALMOST_ FULL	Value Of The Receive FIFO Almost Full Threshold Value, in 64-bit words, of the receive FIFO almost full threshold.

42.4.26 Transmit FIFO Section Empty Threshold (ENETx_TSEM)

Address: Base address + 1A0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								TX_SECTION_EMPTY							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_TSEM field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 TX_SECTION_ EMPTY	Value Of The Transmit FIFO Section Empty Threshold Value, in 64-bit words, of the transmit FIFO section empty threshold.

42.4.27 Transmit FIFO Almost Empty Threshold (ENETx_TAEM)

Address: Base address + 1A4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								TX_ALMOST_EMPTY							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

ENETx_TAEM field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 TX_ALMOST_ EMPTY	Value of Transmit FIFO Almost Empty Threshold Value, in 64-bit words, of the transmit FIFO almost empty threshold.

42.4.28 Transmit FIFO Almost Full Threshold (ENETx_TAFL)

Address: Base address + 1A8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								TX_ALMOST_FULL							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

ENETx_TAFL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 TX_ALMOST_ FULL	Value Of The Transmit FIFO Almost Full Threshold Value, in 64-bit words, of the transmit FIFO almost full threshold. A minimum value of six is required . A recommended value of at least 8 should be set allowing a latency of two clock cycles to the application. If more latency is required the value can be increased as necessary (latency = TAFL - 5). **NOTE:** A FIFO overflow is a fatal error and requires a global reset on the transmit datapath or at least deassertion of ETHEREN.

42.4.29 Transmit Inter-Packet Gap (ENETx_TIPG)

Address: Base address + 1ACh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											IPG				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

ENETx_TIPG field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 IPG	Transmit Inter-Packet Gap Indicates the IPG, in bytes, between transmitted frames. Valid values range from 8 to 27. If value is less than 8, the IPG is 8. If value is greater than 27, the IPG is 27.

42.4.30 Frame Truncation Length (ENETx_FTRL)

Address: Base address + 1B0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																		TRUNC_FL													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

ENETx_FTRL field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–0 TRUNC_FL	Frame Truncation Length

Table continues on the next page...

ENETx_FTRL field descriptions (continued)

Field	Description
	Indicates the value a receive frame is truncated, if it is greater than this value. Must be greater than or equal to RCR[MAX_FL]. **NOTE:** Truncation happens at TRUNC_FL. However, when truncation occurs, the application (FIFO) may receive less data, guaranteeing that it never receives more than the set limit.

42.4.31 Transmit Accelerator Function Configuration (ENETx_TACC)

TACC controls accelerator actions when sending frames. The register can be changed before or after each frame, but it must remain unmodified during frame writes into the transmit FIFO.

The TFWR[STRFWD] field must be set to use the checksum feature.

Address: Base address + 1C0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											PROCHK	IPCHK	0		SHIFT16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_TACC field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 PROCHK	Enables insertion of protocol checksum. 0 Checksum not inserted. 1 If an IP frame with a known protocol is transmitted, the checksum is inserted automatically into the frame. The checksum field must be cleared. The other frames are not modified.
3 IPCHK	Enables insertion of IP header checksum. 0 Checksum is not inserted. 1 If an IP frame is transmitted, the checksum is inserted automatically. The IP header checksum field must be cleared. If a non-IP frame is transmitted the frame is not modified.
2–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ENETx_TACC field descriptions (continued)

Field	Description
0 SHIFT16	TX FIFO Shift-16 0 Disabled. 1 Indicates to the transmit data FIFO that the written frames contain two additional octets before the frame data. This means the actual frame begins at bit 16 of the first word written into the FIFO. This function allows putting the frame payload on a 32-bit boundary in memory, as the 14-byte Ethernet header is extended to a 16-byte header.

42.4.32 Receive Accelerator Function Configuration (ENETx_RACC)

Address: Base address + 1C4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SHIFT16	LINEDIS	0			PRODIS	IPDIS	PADREM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_RACC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 SHIFT16	RX FIFO Shift-16 When this field is set, the actual frame data starts at bit 16 of the first word read from the RX FIFO aligning the Ethernet payload on a 32-bit boundary. **NOTE:** This function only affects the FIFO storage and has no influence on the statistics, which use the actual length of the frame received. 0 Disabled. 1 Instructs the MAC to write two additional bytes in front of each frame received into the RX FIFO.
6 LINEDIS	Enable Discard Of Frames With MAC Layer Errors 0 Frames with errors are not discarded. 1 Any frame received with a CRC, length, or PHY error is automatically discarded and not forwarded to the user application interface.
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ENETx_RACC field descriptions (continued)

Field	Description
2 PRODIS	Enable Discard Of Frames With Wrong Protocol Checksum 0 Frames with wrong checksum are not discarded. 1 If a TCP/IP, UDP/IP, or ICMP/IP frame is received that has a wrong TCP, UDP, or ICMP checksum, the frame is discarded. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
1 IPDIS	Enable Discard Of Frames With Wrong IPv4 Header Checksum 0 Frames with wrong IPv4 header checksum are not discarded. 1 If an IPv4 frame is received with a mismatching header checksum, the frame is discarded. IPv6 has no header checksum and is not affected by this setting. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
0 PADREM	Enable Padding Removal For Short IP Frames 0 Padding not removed. 1 Any bytes following the IP payload section of the frame are removed from the frame.

42.4.33 Timer Control Register (ENETx_ATCR)

ATCR command fields can trigger the corresponding events directly. It is not necessary to preserve any of the configuration fields when a command field is set in the register, that is, no read-modify-write is required. The fields are automatically cleared after the command completes.

Address: Base address + 400h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		SLAVE	0	CAPTURE	0	RESTART	0	PINPER	0		PEREN	OFFRST	OFFEN	0	EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_ATCR field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ENETx_ATCR field descriptions (continued)

Field	Description
13 SLAVE	Enable Timer Slave Mode 0 The timer is active and all configuration fields in this register are relevant. 1 The internal timer is disabled and the externally provided timer value is used. All other fields, except CAPTURE, in this register have no effect. CAPTURE can still be used to capture the current timer value.
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 CAPTURE	Capture Timer Value 0 No effect. 1 The current time is captured and can be read from the ATVR register.
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 RESTART	Reset Timer Resets the timer to zero. This has no effect on the counter enable. If the counter is enabled when this field is set, the timer is reset to zero and starts counting from there. When set, all other fields are ignored during a write.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 PINPER	Enables event signal output assertion on period event. **NOTE:** Not all devices contain the event signal output. See the chip configuration details. 0 Disable. 1 Enable.
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 PEREN	Enable Periodical Event 0 Disable. 1 A period event interrupt can be generated (EIR[TS_TIMER]) and the event signal output is asserted when the timer wraps around according to the periodic setting ATPER. The timer period value must be set before setting this bit. **NOTE:** Not all devices contain the event signal output. See the chip configuration details.
3 OFFRST	Reset Timer On Offset Event 0 The timer is not affected and no action occurs, besides clearing OFFEN, when the offset is reached. 1 If OFFEN is set, the timer resets to zero when the offset setting is reached. The offset event does not cause a timer interrupt.
2 OFFEN	Enable One-Shot Offset Event 0 Disable. 1 The timer can be reset to zero when the given offset time is reached (offset event). The field is cleared when the offset event is reached, so no further event occurs until the field is set again. The timer offset value must be set before setting this field.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ENETx_ATCR field descriptions (continued)

Field	Description
0 EN	Enable Timer 0 The timer stops at the current value. 1 The timer starts incrementing.

42.4.34 Timer Value Register (ENETx_ATVR)

Address: Base address + 404h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATIME																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_ATVR field descriptions

Field	Description
31–0 ATIME	A write sets the timer. A read returns the last captured value. To read the current value, issue a capture command (set ATCR[CAPTURE]) prior to reading this register.

42.4.35 Timer Offset Register (ENETx_ATOFF)

Address: Base address + 408h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OFFSET																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_ATOFF field descriptions

Field	Description
31–0 OFFSET	Offset value for one-shot event generation. When the timer reaches the value, an event can be generated to reset the counter. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds.

42.4.36 Timer Period Register (ENETx_ATPER)

Address: Base address + 40Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERIOD																															
Reset	0	0	1	1	1	0	1	1	1	0	0	1	1	0	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

ENETx_ATPER field descriptions

Field	Description
31–0 PERIOD	Value for generating periodic events. Each instance the timer reaches this value, the period event occurs and the timer restarts. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds. The value should be initialized to 1,000,000,000 (1×10^{9}) to represent a timer wrap around of one second. The increment value set in ATINC should be set to the true nanoseconds of the period of clock ts_clk, hence implementing a true 1 second counter.

42.4.37 Timer Correction Register (ENETx_ATCOR)

Address: Base address + 410h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	COR														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_ATCOR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–0 COR	Correction Counter Wrap-Around Value Defines after how many timer clock cycles (ts_clk) the correction counter should be reset and trigger a correction increment on the timer. The amount of correction is defined in ATINC[INC_CORR]. A value of 0 disables the correction counter and no corrections occur. **NOTE:** This value is given in clock cycles, not in nanoseconds as all other values.

42.4.38 Time-Stamping Clock Period Register (ENETx_ATINC)

Address: Base address + 414h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	INC_CORR							0	INC						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_ATINC field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 INC_CORR	Correction Increment Value This value is added every time the correction timer expires (every clock cycle given in ATCOR). A value smaller than INC slows the timer, while a value larger than INC speeds the timer.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 INC	Clock Period Of The Timestamping Clock (ts_clk) In Nanoseconds The timer increments by this amount each clock cycle. For example, set to 10 for 100 MHz, 8 for 125 MHz, 5 for 200 MHz. **NOTE:** For highest precision, use a value that is an integer fraction of the period set in ATPER.

42.4.39 Timestamp of Last Transmitted Frame (ENETx_ATSTMP)

Address: Base address + 418h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TIMESTAMP																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_ATSTMP field descriptions

Field	Description
31–0 TIMESTAMP	Timestamp of the last frame transmitted by the core that had TxBD[TS] set . This register is only valid when EIR[TS_AVAIL] is set.

42.4.40 Timer Global Status Register (ENETx_TGSR)

Address: Base address + 604h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TF3	TF2	TF1	TF0
W													w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_TGSR field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 TF3	Copy Of Timer Flag For Channel 3 0 Timer Flag for Channel 3 is clear 1 Timer Flag for Channel 3 is set
2 TF2	Copy Of Timer Flag For Channel 2 0 Timer Flag for Channel 2 is clear 1 Timer Flag for Channel 2 is set
1 TF1	Copy Of Timer Flag For Channel 1 0 Timer Flag for Channel 1 is clear 1 Timer Flag for Channel 1 is set
0 TF0	Copy Of Timer Flag For Channel 0 0 Timer Flag for Channel 0 is clear 1 Timer Flag for Channel 0 is set

42.4.41 Timer Control Status Register (ENETx_TCSR*n*)

Address: Base address + 608h offset + (8d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TF	TIE	TMODE				0	TDRE
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_TCSR*n* field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TF	Timer Flag Sets when input capture or output compare occurs. This flag is double buffered between the module clock and 1588 clock domains. When this field is 1, it can be cleared to 0 by writing 1to it. 0 Input Capture or Output Compare has not occurred 1 Input Capture or Output Compare has occurred

Table continues on the next page...

ENETx_TCSR*n* field descriptions (continued)

Field	Description
6 TIE	Timer Interrupt Enable 0 Interrupt is disabled 1 Interrupt is enabled
5–2 TMODE	Timer Mode Updating the Timer Mode field takes a few cycles to register because it is synchronized to the 1588 clock. The version of Timer Mode returned on a read is from the 1588 clock domain. When changing Timer Mode, always disable the channel and read this register to verify the channel is disabled first. 0000 Timer Channel is disabled. 0001 Timer Channel is configured for Input Capture on rising edge 0010 Timer Channel is configured for Input Capture on falling edge 0011 Timer Channel is configured for Input Capture on both edges 0100 Timer Channel is configured for Output Compare - software only 0101 Timer Channel is configured for Output Compare - toggle output on compare 0110 Timer Channel is configured for Output Compare - clear output on compare 0111 Timer Channel is configured for Output Compare - set output on compare 1000 Reserved 1010 Timer Channel is configured for Output Compare - clear output on compare, set output on overflow 10x1 Timer Channel is configured for Output Compare - set output on compare, clear output on overflow 1100 Reserved 1110 Timer Channel is configured for Output Compare - pulse output low on compare for one 1588 clock cycle 1111 Timer Channel is configured for Output Compare - pulse output high on compare for one 1588 clock cycle
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 TDRE	Timer DMA Request Enable 0 DMA request is disabled 1 DMA request is enabled

42.4.42 Timer Compare Capture Register (ENETx_TCCR*n*)

Address: Base address + 60Ch offset + (8d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCC																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENETx_TCCR*n* field descriptions

Field	Description
31–0 TCC	Timer Capture Compare This register is double buffered between the module clock and 1588 clock domains.

ENETx_TCCRn field descriptions (continued)

Field	Description
	When configured for compare, the 1588 clock domain updates with the value in the module clock domain whenever the Timer Channel is first enabled and on each subsequent compare. Write to this register with the first compare value before enabling the Timer Channel. When the Timer Channel is enabled, write the second compare value either immediately, or at least before the first compare occurs. After each compare, write the next compare value before the previous compare occurs and before clearing the Timer Flag. The compare occurs one 1588 clock cycle after the IEEE 1588 Counter increments past the compare value in the 1588 clock domain. If the compare value is less than the value of the 1588 Counter when the Timer Channel is first enabled, then the compare does not occur until following the next overflow of the 1588 Counter. If the compare value is greater than the IEEE 1588 Counter when the 1588 Counter overflows, or the compare value is less than the value of the IEEE 1588 Counter after the overflow, then the compare occurs one 1588 clock cycle following the overflow. When configured for Capture, the value of the IEEE 1588 Counter is captured into the 1588 clock domain and then updated into the module clock domain, provided the Timer Flag is clear. Always read the capture value before clearing the Timer Flag.

42.4.127 Statistic event counters

The following table shows the locations of the statistic event counters in the module's memory map. Definitions of these registers can be found in IETF RFC 2819, *Remote Network Monitoring Management Information Base*.

NOTE

All counters are 32 bits wide with the top 16 bits reserved/ignored, except for the following:

- RMON_T_OCTETS
- IEEE_T_OCTETS_OK
- RMON_R_OCTETS
- IEEE_R_OCTETS_OK

Table 42-158. Statistic event counters memory map

Address offset from ENET base address	Register
0x200	Count of frames not counted correctly (RMON_T_DROP). **NOTE:** Counter not implemented because it is not applicable (read 0 always).
0x204	RMON Tx packet count (RMON_T_PACKETS)
0x208	RMON Tx Broadcast Packets (RMON_T_BC_PKT)
0x20C	RMON Tx Multicast Packets (RMON_T_MC_PKT)
0x210	RMON Tx Packets w CRC/Align error (RMON_T_CRC_ALIGN)
0x214	RMON Tx Packets < 64 bytes, good CRC (RMON_T_UNDERSIZE)

Table continues on the next page...

Table 42-158. Statistic event counters memory map (continued)

Address offset from ENET base address	Register
0x218	RMON Tx Packets > MAX_FL bytes, good CRC (RMON_T_OVERSIZE)
0x21C	RMON Tx Packets < 64 bytes, bad CRC (RMON_T_FRAG)
0x220	RMON Tx Packets > MAX_FL bytes, bad CRC (RMON_T_JAB)
0x224	RMON Tx collision count (RMON_T_COL)
0x228	RMON Tx 64 byte packets (RMON_T_P64)
0x22C	RMON Tx 65 to 127 byte packets (RMON_T_P65TO127n)
0x230	RMON Tx 128 to 255 byte packets (RMON_T_P128TO255n)
0x234	RMON Tx 256 to 511 byte packets (RMON_T_P256TO511)
0x238	RMON Tx 512 to 1023 byte packets (RMON_T_P512TO1023)
0x23C	RMON Tx 1024 to 2047 byte packets (RMON_T_P1024TO2047)
0x240	RMON Tx packets w > 2048 bytes (RMON_T_P_GTE2048)
0x244	RMON Tx Octets (RMON_T_OCTETS)
0x248	Count of frames not counted correctly (IEEE_T_DROP). **NOTE:** Counter not implemented because it is not applicable (read 0 always).
0x24C	Frames Transmitted OK (IEEE_T_FRAME_OK)
0x250	Frames Transmitted with Single Collision (IEEE_T_1COL)
0x254	Frames Transmitted with Multiple Collisions (IEEE_T_MCOL)
0x258	Frames Transmitted after Deferral Delay (IEEE_T_DEF)
0x25C	Frames Transmitted with Late Collision (IEEE_T_LCOL)
0x260	Frames Transmitted with Excessive Collisions (IEEE_T_EXCOL)
0x264	Frames Transmitted with Tx FIFO Underrun (IEEE_T_MACERR)
0x268	Frames Transmitted with Carrier Sense Error (IEEE_T_CSERR)
0x26C	Frames Transmitted with SQE Error (IEEE_T_SQE). **NOTE:** Counter not implemented because there is no SQE information available (read 0 always).
0x270	Flow Control Pause frames transmitted (IEEE_T_FDXFC)
0x274	Octet count for Frames Transmitted w/o Error (IEEE_T_OCTETS_OK). **NOTE:** Counts total octets (includes header and FCS fields).
0x284	RMON Rx packet count (RMON_R_PACKETS)
0x288	RMON Rx Broadcast Packets (RMON_R_BC_PKT)
0x28C	RMON Rx Multicast Packets (RMON_R_MC_PKT)
0x290	RMON Rx Packets w CRC/Align error (RMON_R_CRC_ALIGN)

Table continues on the next page...

Table 42-158. Statistic event counters memory map (continued)

Address offset from ENET base address	Register
0x294	RMON Rx Packets < 64 bytes, good CRC (RMON_R_UNDERSIZE)
0x298	RMON Rx Packets > MAX_FL, good CRC (RMON_R_OVERSIZE)
0x29C	RMON Rx Packets < 64 bytes, bad CRC (RMON_R_FRAG)
0x2A0	RMON Rx Packets > MAX_FL bytes, bad CRC (RMON_R_JAB)
0x2A4	Reserved (RMON_R_RESVD_0)
0x2A8	RMON Rx 64 byte packets (RMON_R_P64)
0x2AC	RMON Rx 65 to 127 byte packets (RMON_R_P65TO127)
0x2B0	RMON Rx 128 to 255 byte packets (RMON_R_P128TO255)
0x2B4	RMON Rx 256 to 511 byte packets (RMON_R_P256TO511)
0x2B8	RMON Rx 512 to 1023 byte packets (RMON_R_P512TO1023)
0x2BC	RMON Rx 1024 to 2047 byte packets (RMON_R_P1024TO2047)
0x2C0	RMON Rx packets w > 2048 bytes (RMON_R_P_GTE2048)
0x2C4	RMON Rx Octets (RMON_R_OCTETS)
0x2C8	Count of frames not counted correctly (IEEE_R_DROP). **NOTE:** Counter increments if a frame with invalid/missing SFD character is detected and has been dropped. None of the other counters increments if this counter increments.
0x2CC	Frames Received OK (IEEE_R_FRAME_OK)
0x2D0	Frames Received with CRC Error (IEEE_R_CRC)
0x2D4	Frames Received with Alignment Error (IEEE_R_ALIGN)
0x2D7	Receive FIFO Overflow count (IEEE_R_MACERR)
0x2DC	Flow Control Pause frames received (IEEE_R_FDXFC)
0x2E0	Octet count for Frames Rcvd w/o Error (IEEE_R_OCTETS_OK). Counts total octets (includes header and FCS fields)

42.5 Functional description

This section provides a complete functional description of the MAC-NET core.

42.5.1 Ethernet MAC frame formats

The IEEE 802.3 standard defines the Ethernet frame format as follows:

- Minimum length of 64 bytes
- Maximum length of 1518 bytes excluding the preamble and the start frame delimiter (SFD) bytes

An Ethernet frame consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Two address fields
- Length or type field
- Data field
- Frame check sequence (CRC value)

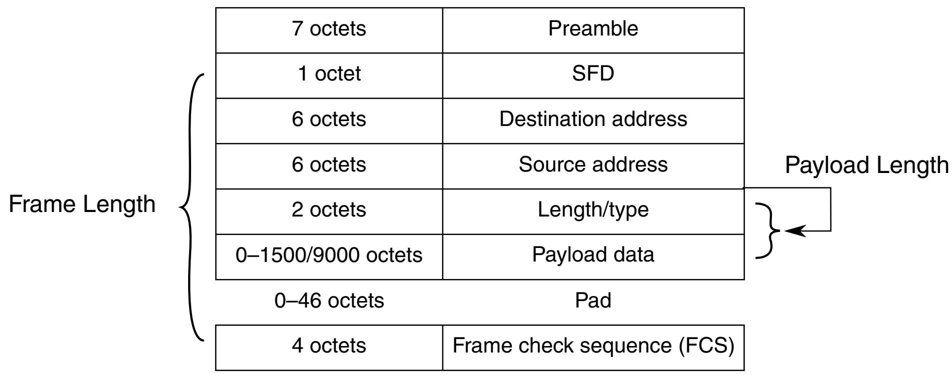


Figure 42-152. MAC frame format overview

Optionally, MAC frames can be VLAN-tagged with an additional four-byte field inserted between the MAC source address and the type/length field. VLAN tagging is defined by the IEEE P802.1q specification. VLAN-tagged frames have a maximum length of 1522 bytes, excluding the preamble and the SFD bytes.

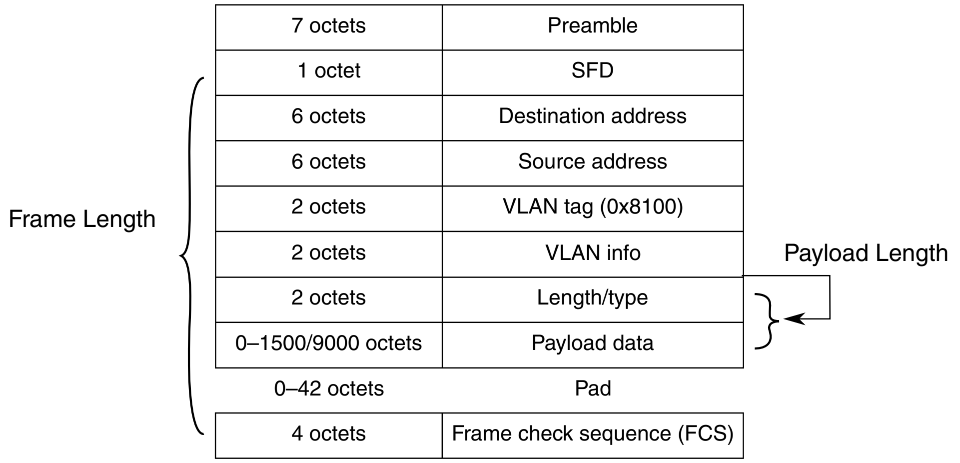


Figure 42-153. VLAN-tagged MAC frame format overview

Table 42-159. MAC frame definition

Term	Description
Frame length	Defines the length, in octets, of the complete frame without preamble and SFD. A frame has a valid length if it contains at least 64 octets and does not exceed the programmed maximum length.
Payload length	The length/type field indicates the length of the frame's payload section. The most significant byte is sent/received first. • If the length/type field is set to a value less than 46, the payload is padded so that the minimum frame length requirement (64 bytes) is met. For VLAN-tagged frames, a value less than 42 indicates a padded frame. • If the length/type field is set to a value larger than the programmed frame maximum length (e.g. 1518) it is interpreted as a type field.
Destination and source address	48-bit MAC addresses. The least significant byte is sent/received first and the first two least significant bits of the MAC address distinguish MAC frames, as detailed in MAC address check.

Note

Although the IEEE specification defines a maximum frame length, the MAC core provides the flexibility to program any value for the frame maximum length.

42.5.1.1 Pause Frames

The receiving device generates a pause frame to indicate a congestion to the emitting device, which should stop sending data.

Pause frames are indicated by the length/type set to 0x8808. The two first bytes of a pause frame following the type, defines a 16-bit opcode field set to 0x0001 always. A 16-bit pause quanta is defined in the frame payload bytes 2 (P1) and 3 (P2) as defined in the following table. The P1 pause quanta byte is the most significant.

Table 42-160. Pause Frame Format (Values in Hex)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					
15	16	17	18	19	20	21	22	23	24	25	26	27 –68	
00	00	00	00	00	00	88	08	00	01	hi	lo	00	
Source Address						Type		Opcode		P1	P2	pad (42)	
69	70	71	72										
26	6B	AE	0A										
CRC-32													

There is no payload length field found within a pause frame and a pause frame is always padded with 42 bytes (0x00).

If a pause frame with a pause value greater than zero (XOFF condition) is received, the MAC stops transmitting data as soon the current frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One pause quanta fraction refers to 512 bit times.

If a pause frame with a pause value of zero (XON condition) is received, the transmitter is allowed to send data immediately (see Full-duplex flow control operation for details).

42.5.1.2 Magic packets

A magic packet is a unicast, multicast, or broadcast packet, which carries a defined sequence in the payload section.

Magic packets are received and inspected only under specific conditions as described in Magic packet detection.

The defined sequence to decode a magic packet is formed with a synchronization stream which consists of six consecutive 0xFF bytes, and is followed by sequence of sixteen consecutive unicast MAC addresses of the node to be awakened.

This sequence can be located anywhere in the magic packet payload. The magic packet is formed with a standard Ethernet header, optional padding, and CRC.

42.5.2 IP and higher layers frame format

The following sections use the term datagram to describe the protocol specific data unit that is found within the payload section of its container entity.

For example, an IP datagram specifies the payload section of an Ethernet frame. A TCP datagram specifies the payload section within an IP datagram.

42.5.2.1 Ethernet types

IP datagrams are carried in the payload section of an Ethernet frame. The Ethernet frame type/length field discriminates several datagram types.

The following table lists the types of interest:

Table 42-161. Ethernet type value examples

Type	Description
0x8100	VLAN-tagged frame. The actual type is found 4 octets later in the frame.
0x0800	IPv4
0x0806	ARP
0x86DD	IPv6

42.5.2.2 IPv4 datagram format

The following figure shows the IP Version 4 (IPv4) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words. The first byte sent/received is the leftmost byte of the first word (in other words, version/IHL field).

The IP header can contain further options, which are always padded if necessary to guarantee the payload following the header is aligned to a 32-bit boundary.

The IP header is immediately followed by the payload, which can contain further protocol headers (for example, TCP or UDP, as indicated by the protocol field value). The complete IP datagram is transported in the payload section of an Ethernet frame.

Table 42-162. IPv4 header format

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="8">TOS</td><td colspan="16">Length</td></tr>
<tr><td colspan="16">Fragment ID</td><td colspan="3">Flags</td><td colspan="13">Fragment offset</td></tr>
<tr><td colspan="8">TTL</td><td colspan="8">Protocol</td><td colspan="16">Header checksum</td></tr>
<tr><td colspan="32">Source address</td></tr>
<tr><td colspan="32">Destination address</td></tr>
<tr><td colspan="32">Options</td></tr>
</table>

Table 42-163. IPv4 header fields

Field name	Description
Version	4-bit IP version information. 0x4 for IPv4 frames.
IHL	4-bit Internet header length information. Determines number of 32-bit words found within the IP header. If no options are present, the default value is 0x5.
TOS	Type of service/DiffServ field.
Length	Total length of the datagram in bytes, including all octets of header and payload.
Fragment ID, flags, fragment offset	Fields used for IP fragmentation.
TTL	Time-to-live. In effect, is decremented at each router arrival. If zero, datagram must be discarded.
Protocol	Identifier of protocol that follows in the datagram.
Header checksum	Checksum of IP header. For computational purposes, this field's value is zero.
Source address	Source IP address.
Destination address	Destination IP address.

42.5.2.3 IPv6 datagram format

The following figure shows the IP version 6 (IPv6) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words and has a fixed length of ten words (40 bytes). The next header field identifies the type of the header that follows the IPv6 header. It is defined similar to the protocol identifier within IPv4, with new definitions for identifying extension headers. These headers can be inserted between the IPv6 header and the protocol header, which will shift the protocol header accordingly.The accelerator currently only supports IPv6 without extension headers (in other words, the next header specifies TCP, UDP, or IMCP).

The first byte sent/received is the leftmost byte of the first word (in other words, version/ traffic class fields).

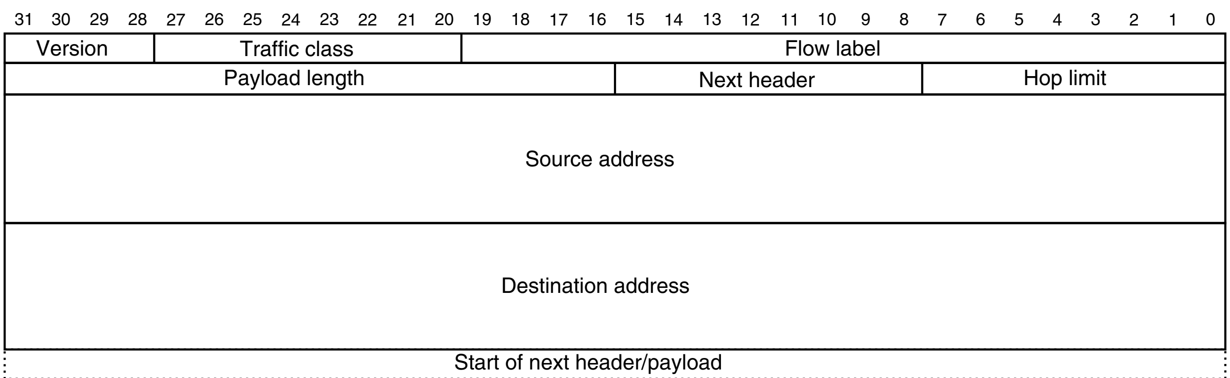


Figure 42-154. IPv6 header format

Table 42-164. IPv6 header fields

Field name	Description
Version	4-bit IP version information. 0x6 for all IPv6 frames.
Traffic class	8-bit field defining the traffic class.
Flow label	20-bit flow label identifying frames of the same flow.
Payload length	16-bit length of the datagram payload in bytes. It includes all octets following the IPv6 header.
Next header	Identifies the header that follows the IPv6 header. This can be the protocol header or any IPv6 defined extension header.
Hop limit	Hop counter, decremented by one by each station that forwards the frame. If hop limit is 0 the frame must be discarded.
Source address	128-bit IPv6 source address.
Destination address	128-bit IPv6 destination address.

42.5.2.4 Internet Control Message Protocol (ICMP) datagram format

An internet control message protocol (ICMP) is found following the IP header, if the protocol identifier is 1. The ICMP datagram has a four-octet header followed by additional message data.

Table 42-165. ICMP header format

<table>
<tr><td>31</td><td>30 29 28</td><td>27 26 25 24</td><td>23 22 21 20</td><td>19 18 17 16</td><td>15 14 13 12</td><td>11 10 9 8</td><td>7 6 5 4</td><td>3 2 1 0</td></tr>
<tr><td colspan="3">Type</td><td colspan="2">Code</td><td colspan="4">Checksum</td></tr>
<tr><td colspan="9">ICMP message data</td></tr>
</table>

Table 42-166. IP header fields

Field name	Description
Type	8-bit type information
Code	8-bit code that is related to the message type
Checksum	16-bit one's complement checksum over the complete ICMP datagram

42.5.2.5 User Datagram Protocol (UDP) datagram format

A user datagram protocol header is found after the IP header, when the protocol identifier is 17.

The payload of the datagram is after the UDP header. The header byte order follows the conventions given for the IP header above.

Table 42-167. UDP header format

31	30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port					Destination port			
Length					Checksum			

Table 42-168. UDP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Length	Length of user data which immediately follows the header, including the UDP header (that is, minimum value is 8)
Checksum	Checksum over the complete datagram and some IP header information

42.5.2.6 TCP datagram format

A TCP header is found following the IP header, when the protocol identifier has a value of 6.

The TCP payload immediately follows the TCP header.

Table 42-169. TCP header format

31	30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port					Destination port			
Sequence number								
Acknowledgement number								
Offset				Flags	Window			
Checksum					Urgent pointer			
Options								

Table 42-170. TCP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Sequence number	Transmit sequence number
Ack. number	Receive sequence number
Offset	Data offset, which is number of 32-bit words within TCP header — if no options selected, defaults to value of 5

Table continues on the next page...

Table 42-170. TCP header fields (continued)

Field name	Description
Flags	URG, ACK, PSH, RST, SYN, FIN flags
Window	TCP receive window size information
Checksum	Checksum over the complete datagram (TCP header and data) and IP header information
Options	Additional 32-bit words for protocol options

42.5.3 IEEE 1588 message formats

The following sections describe the IEEE 1588 message formats.

42.5.3.1 Transport encapsulation

The precision time protocol (PTP) datagrams are encapsulated in Ethernet frames using the UDP/IP transport mechanism, or optionally, with the newer 1588v2 directly in Ethernet frames (layer 2).

Typically, multicast addresses are used to allow efficient distribution of the synchronization messages.

42.5.3.1.1 UDP/IP

The 1588 messages (v1 and v2) can be transported using UDP/IP multicast messages.

Table 42-171 shows IP multicast groups defined for PTP. The table also shows their respective MAC layer multicast address mapping according to RFC 1112 (last three octets of IP follow the fixed value of 01-00-5E).

Table 42-171. UDP/IP multicast domains

Name	IP Address	MAC Address mapping
DefaultPTPdomain	224.0.1.129	01-00-5E-00-01-81
AlternatePTPdomain1	224.0.1.130	01-00-5E-00-01-82
AlternatePTPdomain2	224.0.1.131	01-00-5E-00-01-83
AlternatePTPdomain3	224.0.1.132	01-00-5E-00-01-84

Table 42-172. UDP port numbers

Message type	UDP port	Note
Event	319	Used for SYNC and DELAY_REQUEST messages
General	320	All other messages (for example, follow-up, delay-response)

42.5.3.1.2 Native Ethernet (PTPv2)

In addition to using UDP/IP frames, IEEE 1588v2 defines a native Ethernet frame format that uses ethertype = 0x88F7. The payload of the Ethernet frame immediately contains the PTP datagram, starting with the PTPv2 header.

Besides others, version 2 adds a peer delay mechanism to allow delay measurements between individual point-to-point links along a path over multiple nodes. The following multicast domains are also defined in PTPv2.

Table 42-173. PTPv2 multicast domains

Name	MAC address
Normal messages	01-1B-19-00-00-00
Peer delay messages	01-80-C2-00-00-0E

42.5.3.2 PTP header

All PTP frames contain a common header that determines the protocol version and the type of message, which defines the remaining content of the message.

All multi-octet fields are transmitted in big-endian order (the most significant byte is transmitted/received first).

The last four bits of versionPTP are at the same position (second byte) for PTPv1 and PTPv2 headers. This allows accurate identification by inspecting the first two bytes of the message.

42.5.3.2.1 PTPv1 header

Table 42-174. Common PTPv1 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	2	versionPTP = 0x0001							
2	2	versionNetwork							
4	16	subdomain							
20	1	messageType							
21	1	sourceCommunicationTechnology							
22	6	sourceUuid							
28	2	sourcePortId							
30	2	sequenceId							

Table continues on the next page...

Table 42-174. Common PTPv1 message header (continued)

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
32	1	control							
33	1	0x00							
34	2	flags							
36	4	reserved							

The type of message is encoded in the messageType and control fields as shown in Table 42-175 :

Table 42-175. PTPv1 message type identification

messageType	control	Message Name	Message
0x01	0x0	SYNC	Event message
0x01	0x1	DELAY_REQ	Event message
0x02	0x2	FOLLOW_UP	General message
0x02	0x3	DELAY_RESP	General message
0x02	0x4	MANAGEMENT	General message
other	other	—	Reserved

The field sequenceId is used to non-ambiguously identify a message.

42.5.3.2.2 PTPv2 header

Table 42-176. Common PTPv2 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	1	transportSpecific				messageId			
1	1	reserved				versionPTP = 0x2			
2	2	messageLength							
4	1	domainNumber							
5	1	reserved							
6	2	flags							
8	8	correctionField							
16	4	reserved							
20	10	sourcePortIdentity							
30	2	sequenceId							
32	1	control							
33	1	logMeanMessageInterval							

The type of message is encoded in the field messageId as follows:

Table 42-177. PTPv2 message type identification

messageId	Message name	Message
0x0	SYNC	Event message
0x1	DELAY_REQ	Event message
0x2	PATH_DELAY_REQ	Event message
0x3	PATH_DELAY_RESP	Event message
0x4–0x7	—	Reserved
0x8	FOLLOW_UP	General message
0x9	DELAY_RESP	General message
0xa	PATH_DELAY_FOLLOW_UP	General message
0xb	ANNOUNCE	General message
0xc	SIGNALING	General message
0xd	MANAGEMENT	General message

The PTPv2 flags field contains further details on the type of message, especially if one-step or two-step implementations are used. The one- or two-step implementation is controlled by the TWO_STEP bit in the first octet of the flags field as shown below. Reserved bits are cleared.

Table 42-178. PTPv2 message flags field definitions

Bit	Name	Description
0	ALTERNATE_MASTER	See IEEE 1588 Clause 17.4
1	TWO_STEP	1 Two-step clock 0 One-step clock
2	UNICAST	1 Transport layer address uses a unicast destination address 0 Multicast is used
3	—	Reserved
4	—	Reserved
5	Profile specific	
6	Profile specific	
7	—	Reserved

42.5.4 MAC receive

The MAC receive engine performs the following tasks:

- Check frame framing

- Remove frame preamble and frame SFD field
- Discard frame based on frame destination address field
- Terminate pause frames
- Check frame length
- Remove payload padding if it exists
- Calculate and verify CRC-32
- Write received frames in the core receive FIFO

If the MAC is programmed to operate in half-duplex mode, it will also check if the frame is received with a collision.

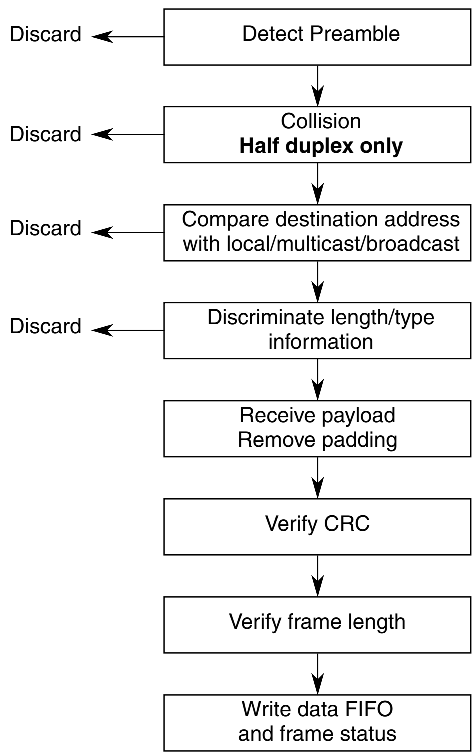


Figure 42-155. MAC receive flow

42.5.4.1 Collision detection in half-duplex mode

If the packet is received with a collision detected during reception of the first 64 bytes, the packet is discarded (if frame size was less than ~14 octets) or transmitted to the user application with an error and RxBD[CE] set.

42.5.4.2 Preamble processing

The IEEE 802.3 standard allows a maximum size of 56 bits (seven bytes) for the preamble, while the MAC core allows any preamble length, including zero length preamble.

The MAC core checks for the start frame delimiter (SFD) byte. If the next byte of the preamble, which is different from 0x55, is not 0xD5, the frame is discarded.

Although the IEEE specification dictates that the inner-packet gap should be at least 96 bits, the MAC core is designed to accept frames separated by only 64 10/100 Mbps operation (MII) bits.

The MAC core removes the preamble and SFD bytes.

42.5.4.3 MAC address check

The destination address bit 0 differentiates between multicast and unicast addresses.

- If bit 0 is 0, the MAC address is an individual (unicast) address.
- If bit 0 is 1, the MAC address defines a group (multicast) address.
- If all 48 bits of the MAC address are set, it indicates a broadcast address.

42.5.4.3.1 Unicast address check

If a unicast address is received, the destination MAC address is compared to the node MAC address programmed by the host in the PADDR1/2 registers.

If the destination address matches any of the programmed MAC addresses, the frame is accepted.

If Promiscuous mode is enabled (RCR[PROM] = 1) no address checking is performed and all unicast frames are accepted.

42.5.4.3.2 Multicast and unicast address resolution

The hash table algorithm used in the group and individual hash filtering operates as follows.

- The 48-bit destination address is mapped into one of 64 bits, represented by 64 bits in ENET*n*_GAUR/GALR (group address hash match) or ENET*n*_IAUR/IALR (individual address hash match).

- This mapping is performed by passing the 48-bit address through the on-chip 32-bit CRC generator and selecting the six most significant bits of the CRC-encoded result to generate a number between 0 and 63.
- The msb of the CRC result selects ENET*n*_GAUR (msb = 1) or ENET*n*_GALR (msb = 0).
- The five lsbs of the hash result select the bit within the selected register.
- If the CRC generator selects a bit set in the hash table, the frame is accepted; else, it is rejected.

For example, if eight group addresses are stored in the hash table and random group addresses are received, the hash table prevents roughly 56/64 (or 87.5%) of the group address frames from reaching memory. Those that do reach memory must be further filtered by the processor to determine if they truly contain one of the eight desired addresses.

The effectiveness of the hash table declines as the number of addresses increases.

The user must initialize the hash table registers. Use this CRC32 polynomial to compute the hash:

- $FCS(x) = x^{32}+ x^{26}+ x^{23}+ x^{22}+ x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^{8}+ x^{7}+ x^{5}+ x^{4}+ x^{2}+ x^{1}+ 1$

If Promiscuous mode is enabled (ENET*n*_RCR[PROM] = 1) all unicast and multicast frames are accepted regardless of ENET*n*_GAUR/GALR and ENET*n*_IAUR/IALR settings.

42.5.4.3.3 Broadcast address reject

All broadcast frames are accepted if BC_REJ is cleared or ENET*n*_RCR[PROM] is set. If PROM is cleared when ENET*n*_RCR[BC_REJ] is set, all broadcast frames are rejected.

Table 42-179. Broadcast address reject programming

PROM	BC_REJ	Broadcast frames
0	0	Accepted
0	1	Rejected
1	0	Accepted
1	1	Accepted

42.5.4.3.4 Miss-bit implementation

For higher layer filtering purposes, RxBD[M] indicates an address miss when the MAC operates in promiscuous mode and accepts a frame that would otherwise be rejected.

If a group/individual hash or exact match does not occur and Promiscuous mode is enabled (RCR[PROM] = 1), the frame is accepted and the M bit is set in the buffer descriptor; otherwise, the frame is rejected.

This means the status bit is set in any of the following conditions during Promiscuous mode:

- A broadcast frame is received when BC_REJ is set
- A unicast is received that does not match either:
  - Node address (PALR[PADDR1] and PAUR[PADDR2])
  - Hash table for unicast (IAUR[IADDR1] and IALR[IADDR2])
- A multicast is received that does not match the GAUR[GADDR1] and GALR[GADDR2] hash table entries

42.5.4.4 Frame length/type verification: payload length check

If the length/type is less than 0x600 and NLC is set, the MAC checks the payload length and reports any error in the frame status word and interrupt bit PLR.

If the length/type is greater than or equal to 0x600, the MAC interprets the field as a type and no payload length check is performed.

The length check is performed on VLAN and stacked VLAN frames. If a padded frame is received, no length check can be performed due to the extended frame payload because padded frames can never have a payload length error.

42.5.4.5 Frame length/type verification: frame length check

When the receive frame length exceeds MAX_FL bytes, the BABR interrupt is generated and the RxBD[LG] bit is set.

The frame is not truncated unless the frame length exceeds the value programmed in ENET*n*_FTRL[TRUNC_FL]. If the frame is truncated, RxBD[TR] is set. In addition, a truncated frame always has the CRC error indication set (RxBD[CR]).

42.5.4.6 VLAN frames processing

VLAN frames have a length/type field set to 0x8100 immediately followed by a 16-Bit VLAN control information field.

VLAN-tagged frames are received as normal frames because the VLAN tag is not interpreted by the MAC function, and are pushed complete with the VLAN tag to the user application. If the length/type field of the VLAN-tagged frame, which is found four octets later in the frame, is less than 42, the padding is removed. In addition, the frame status word (RxBD[NO]) indicates that the current frame is VLAN tagged.

42.5.4.7 Pause frame termination

The receive engine terminates pause frames and does not transfer them to the receive FIFO. The quanta is extracted and sent to the MAC transmit path via a small internal clock rate decoupling asynchronous FIFO.

The quanta is written only if a correct CRC and frame length are detected by the control state machine. If not, the quanta is discarded and the MAC transmit path is not paused.

Good pause frames are ignored if ENET*n*_RCR[FCE] is cleared and are forwarded to the client interface when ENET*n*_RCR[PAUFWD] is set.

42.5.4.8 CRC check

The CRC-32 field is checked and forwarded to the core FIFO interface if ENET*n*_RCR[CRCFWD] is cleared and ENET*n*_RCR[PADEN] is set.

When CRCFWD is set (regardless of PADEN), the CRC-32 field is checked and terminated (not transmitted to the FIFO).

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the frame check sequence (FCS) field with the x^{31} term as right-most bit of the first octet. The CRC bits are thus received in the following order: $x^{31}, x^{30}, \ldots, x^{1}, x^{0}$.

If a CRC error is detected, the frame is marked invalid and RxBD[CR] is set.

42.5.4.9 Frame padding removal

When a frame is received with a payload length field set to less than 46 (42 for VLAN-tagged frames and 38 for frames with stacked VLANs), the zero padding can be removed before the frame is written into the data FIFO depending on the setting of ENET*n*_RCR[PADEN].

Note

If a frame is received with excess padding (in other words, the length field is set as mentioned above, but the frame has more than 64 octets) and padding removal is enabled, then the padding is removed as normal and no error is reported if the frame is otherwise correct (for example: good CRC, less than maximum length, and no other error).

42.5.5 MAC transmit

Frame transmission starts when the transmit FIFO holds enough data.

After a transfer starts, the MAC transmit function performs the following tasks:

- Generates preamble and SFD field before frame transmission
- Generates XOFF pause frames if the receive FIFO reports a congestion or if ENET*n*_TCR[TFC_PAUSE] is set with ENET*n*_OPD[PAUSE_DUR] set to a non-zero value
- Generates XON pause frames if the receive FIFO congestion condition is cleared or if TFC_PAUSE is set with PAUSE_DUR cleared
- Suspends Ethernet frame transfer (XOFF) if a non-zero pause quanta is received from the MAC receive path
- Adds padding to the frame if required
- Calculates and appends CRC-32 to the transmitted frame
- Sends the frame with correct inter-packet gap (IPG) (deferring)

When the MAC is configured to operate in half-duplex mode, the following additional tasks are performed:

- Collision detection
- Frame retransmit after back-off timer expires

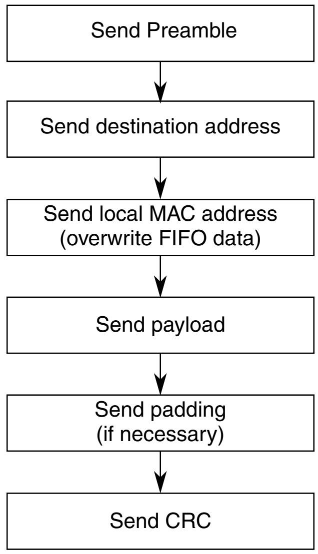


Figure 42-156. Frame transmit overview

42.5.5.1 Frame payload padding

The IEEE specification defines a minimum frame length of 64 bytes.

If the frame sent to the MAC from the user application has a size smaller than 60 bytes, the MAC automatically adds padding bytes (0x00) to comply with the Ethernet minimum frame length specification. Transmit padding is always performed and cannot be disabled.

If the MAC is not allowed to append a CRC (TxBD[TC] = 1), the user application is responsible for providing frames with a minimum length of 64 octets.

42.5.5.2 MAC address insertion

On each frame received from the core transmit FIFO interface, the source MAC address is either:

- Replaced by the address programmed in the PADDR1/2 fields (ENET*n*_TCR[ADDINS] = 1)
- Transparently forwarded to the Ethernet line (ENET*n*_TCR[ADDINS] = 0)

42.5.5.3 CRC-32 generation

The CRC-32 field is optionally generated and appended at the end of a frame.

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the FCS field so that the x^{31} term is the right-most bit of the first octet. The CRC bits are thus transmitted in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

42.5.5.4 Inter-packet gap (IPG)

In full-duplex mode, after frame transmission and before transmission of a new frame, an IPG (programmed in ENET*n*_TIPG) is maintained. The minimum IPG can be programmed between 8 and 27 byte-times (64 and 216 bit-times).

In half-duplex mode, the core constantly monitors the line. Actual transmission of the data onto the network occurs only if it has been idle for a 96-bit time period, and any back-off time requirements have been satisfied. In accordance with the standard, the core begins to measure the IPG from CRS de-assertion.

42.5.5.5 Collision detection and handling — half-duplex operation only

A collision occurs on a half-duplex network when concurrent transmissions from two or more nodes take place. During transmission, the core monitors the line condition and detects a collision when the PHY device asserts COL.

When the core detects a collision while transmitting, it stops transmission of the data and transmits a 32-bit jam pattern. If the collision is detected during the preamble or the SFD transmission, the jam pattern is transmitted after completing the SFD, which results in a minimum 96-bit fragment. The jam pattern is a fixed pattern that is not compared to the actual frame CRC, and has a very low probability (0.532) of having a jam pattern identical to the CRC.

If a collision occurs before transmission of 64 bytes (including preamble and SFD), the MAC core waits for the backoff period and retransmits the packet data (stored in a 64-byte re-transmit buffer) that has already been sent on the line. The backoff period is generated from a pseudo-random process (truncated binary exponential backoff).

If a collision occurs after transmission of 64 bytes (including preamble and SFD), the MAC discards the remainder of the frame, optionally sets the LC interrupt bit, and sets TxBD[LCE].

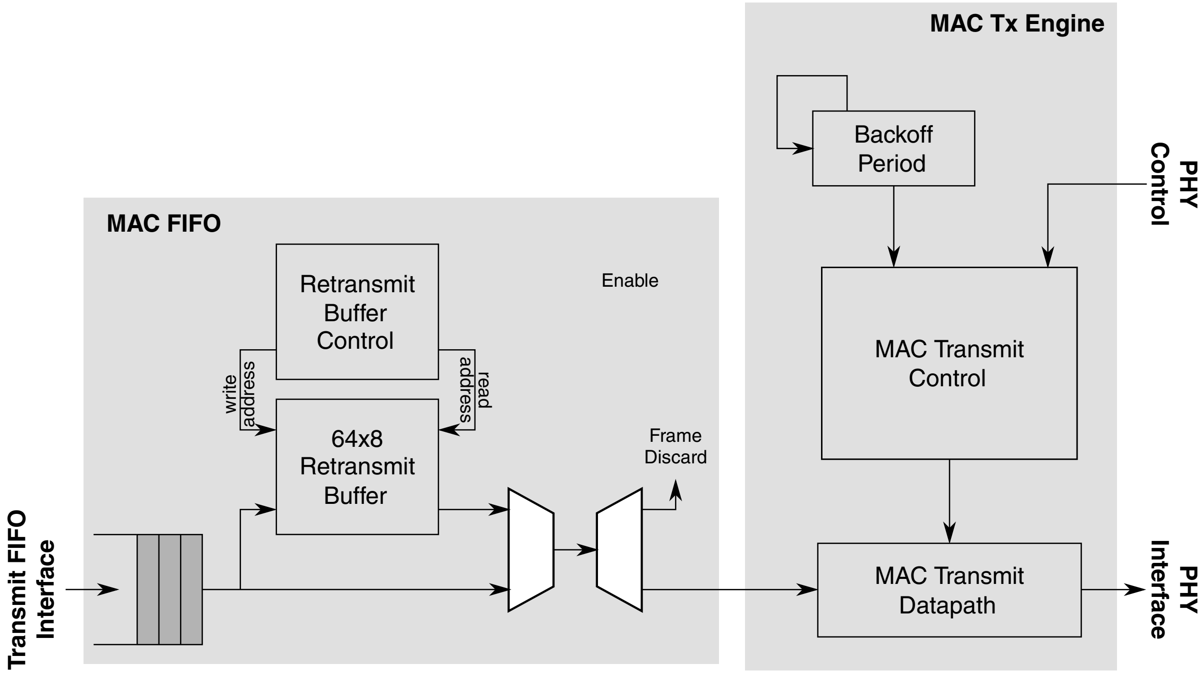


Figure 42-157. Packet re-transmit overview

The backoff time is represented by an integer multiple of slot times. One slot is equal to a 512-bit time period. The number of the delay slot times, before the n^{th} re-transmission attempt, is chosen as a uniformly-distributed random integer in the range:

- $0 < r < 2^k$
- k = min(*n*, N); where *n* is the number of retransmissions and N = 10

For example, after the first collision, the backoff period is 0 or 1 slot time. If a collision occurs on the first retransmission, the backoff period is 0, 1, 2, or 3, and so on.

The maximum backoff time (in 512-bit time slots) is limited by N = 10 as specified in the IEEE 802.3 standard.

If a collision occurs after 16 consecutive retransmissions, the core reports an excessive collision condition (ENET*n*_EIR[RL] interrupt field and TxBD[EE]) and discards the current packet from the FIFO.

In networks violating the standard requirements, a collision may occur after transmission of the first 64 bytes. In this case, the core stops the current packet transmission and discards the rest of the packet from the transmit FIFO. The core resumes transmission with the next packet available in the core transmit FIFO.

42.5.6 Full-duplex flow control operation

Three conditions are handled by the core's flow control engine:

- Remote device congestion — The remote device connected to the same Ethernet segment as the core reports congestion and requests that the core stop sending data.
- Core FIFO congestion — When the core's receive FIFO reaches a user-programmable threshold (RX section empty), the core sends a pause frame back to the remote device requesting the data transfer to stop.
- Local device congestion — Any device connected to the core can request (typically, via the host processor) the remote device to stop transmitting data.

42.5.6.1 Remote device congestion

When the MAC transmit control gets a valid pause quanta from the receive path and if ENET*n*_RCR[FCE] is set, the MAC transmit logic:

- Completes the transfer of the current frame.
- Stops sending data for the amount of time specified by the pause quanta in 512 bit time increments.
- Sets ENET*n*_TCR[RFC_PAUSE].

Frame transfer resumes when the time specified by the quanta expires and if no new quanta value is received, or if a new pause frame with a quanta value set to 0x0000 is received. The MAC also resets RFC_PAUSE to zero.

If ENET*n*_RCR[FCE] cleared, the MAC ignores received pause frames.

Optionally and independent of ENET*n*_RCR[FCE], pause frames are forwarded to the client interface if PAUFWD is set.

42.5.6.2 Local device/FIFO congestion

The MAC transmit engine generates pause frames when the local receive FIFO is not able to receive more than a pre-defined number of words (FIFO programmable threshold) or when pause frame generation is requested by the local host processor.

- To generate a pause frame, the host processor sets ENET*n*_TCR[TFC_PAUSE]. A single pause frame is generated when the current frame transfer is completed and TFC_PAUSE is automatically cleared. Optionally, an interrupt is generated.
- An XOFF pause frame is generated when the receive FIFO asserts its section empty flag (internal). An XOFF pause frame is generated automatically, when the current frame transfer completes.
- An XON pause frame is generated when the receive FIFO deasserts its section empty flag (internal). An XON pause frame is generated automatically, when the current frame transfer completes.

When an XOFF pause frame is generated, the pause quanta (payload byte P1 and P2) is filled with the value programmed in ENET*n*_OPD[PAUSE_DUR].

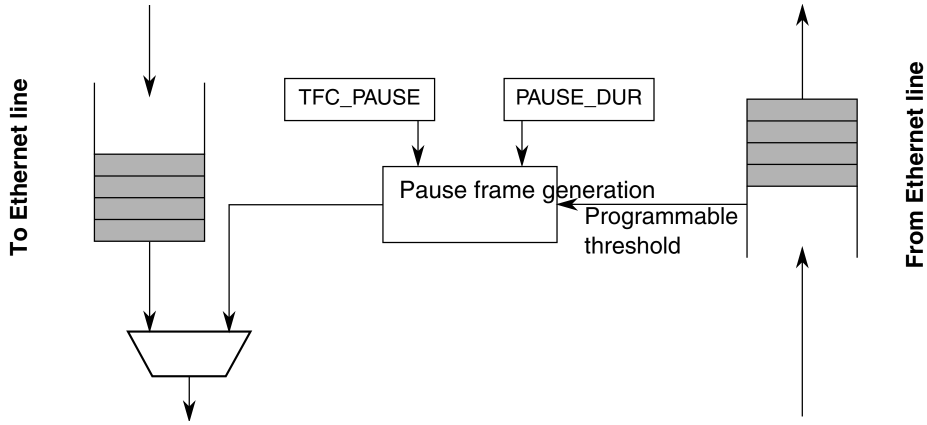


Figure 42-158. Pause frame generation overview

Note

Although the flow control mechanism should prevent any FIFO overflow on the MAC core receive path, the core receive FIFO is protected. When an overflow is detected on the receive FIFO, the current frame is truncated with an error indication set in the frame status word. The frame should subsequently be discarded by the user application.

42.5.7 Magic packet detection

Magic packet detection wakes a node that is put in power-down mode by the node management agent. Magic packet detection is supported only if the MAC is configured in sleep mode.

42.5.7.1 Sleep mode

To put the MAC in Sleep mode, set ENET*n*_ECR[SLEEP]. At the same time ENET*n*_ECR[MAGICEN] should also be set to enable magic packet detection.

In addition, when the processor is in Stop mode, Sleep mode is entered, without affecting the ENET*n*_ECR register bits.

When the core is in Sleep mode:

- The MAC transmit logic is disabled.
- The core FIFO receive/transmit functions are disabled.
- The MAC receive logic is kept in Normal mode, but it ignores all traffic from the line except magic packets. They are detected so that a remote agent can wake the node.

42.5.7.2 Magic packet detection

The core is designed to detect magic packets (see Magic packets) with the destination address set to:

- Any multicast address
- The broadcast address
- The unicast address programmed in PADDR1/2

When a magic packet is detected, EIR[WAKEUP] is set and none of the statistic registers are incremented.

42.5.7.3 Wakeup

When a magic packet is detected, indicated by ENET*n*_EIR[WAKEUP], ENET*n*_ECR[SLEEP] should be cleared to resume normal operation of the MAC. Clearing the SLEEP bit automatically masks ENET*n*_ECR[MAGICEN], disabling magic packet detection.

42.5.8 IP accelerator functions

The following sections describe the IP accelerator functions.

42.5.8.1 Checksum calculation

The IP and ICMP, TCP, UDP checksums are calculated with one's complement arithmetic summing up 16-bit values.

- For ICMP, the checksum is calculated over the complete ICMP datagram, in other words without IP header.
- For TCP and UDP, the checksums contain the header and data sections and values from the IP header, which can be seen as a pseudo-header that is not actually present in the data stream.

Table 42-180. IPv4 pseudo-header for checksum calculation

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">Source address</td></tr>
<tr><td colspan="32">Destination address</td></tr>
<tr><td colspan="8">Zero</td><td colspan="8">Protocol</td><td colspan="16">TCP/UDP length</td></tr>
</table>

Table 42-181. IPv6 pseudo-header for checksum calculation

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="32">Source address</td></tr>
<tr><td colspan="32">Destination address</td></tr>
<tr><td colspan="32">TCP/UDP length</td></tr>
<tr><td colspan="24">Zero</td><td colspan="8">Next header</td></tr>
</table>

The TCP/UDP length value is the length of the TCP or UDP datagram, which is equal to the payload of an IP datagram. It is derived by subtracting the IP header length from the complete IP datagram length that is given in the IP header (IPv4), or directly taken from the IP header (IPv6). The protocol field is the corresponding value from the IP header. The Zero fields are all zeroes.

For IPv6, the complete 128-bit addresses are considered. The next header value identifies the upper layer protocol as either TCP or UDP. It may differ from the next header value of the IPv6 header if extension headers are inserted before the protocol header.

The checksum calculation uses 16-bit words in network byte order: The first byte sent/received is the MSB, and the second byte sent/received is the LSB of the 16-bit value to add to the checksum. If the frame ends on an odd number of bytes, a zero byte is appended for checksum calculation only, and is not actually transmitted.

42.5.8.2 Additional padding processing

According to IEEE 802.3, any Ethernet frame must have a minimum length of 64 octets.

The MAC usually removes padding on receive when a frame with length information is received. Because IP frames have a type value instead of length, the MAC does not remove padding for short IP frames, as it is not aware of the frame contents.

The IP accelerator function can be configured to remove the Ethernet padding bytes that might follow the IP datagram.

On transmit, the MAC automatically adds padding as necessary to fill any frame to a 64-byte length.

42.5.8.3 32-bit Ethernet payload alignment

The data FIFOs allow inserting two additional arbitrary bytes in front of a frame. This extends the 14-byte Ethernet header to a 16-byte header, which leads to alignment of the Ethernet payload, following the Ethernet header, on a 32-bit boundary.

This function can be enabled for transmit and receive independently with the corresponding SHIFT16 bits in the ENET*n*_TACC and ENET*n*_RACC registers.

When enabled, the valid frame data is arranged as shown in Table 42-182.

Table 42-182. 64-bit interface data structure with SHIFT16 enabled

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	Any value	Any value
Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8	Byte 7	Byte 6
...							

42.5.8.3.1 Receive processing

When ENET*n*_RACC[SHIFT16] is set, each frame is received with two additional bytes in front of the frame.

The user application must ignore these first two bytes and find the first byte of the frame in bits 23–16 of the first word from the RX FIFO.

Note

> SHIFT16 must be set during initialization and kept set during the complete operation, because it influences the FIFO write behavior.

42.5.8.3.2 Transmit processing

When ENET*n*_TACC[SHIFT16] is set, the first two bytes of the first word written (bits 15–0) are discarded immediately by the FIFO write logic.

The SHIFT16 bit can be enabled/disabled for each frame individually if required, but can be changed only between frames.

42.5.8.4 Received frame discard

Because the receive FIFO must be operated in store and forward mode (ENET*n*_RSFL cleared), received frames can be discarded based on the following errors:

- The MAC function receives the frame with an error:
  - The frame has an invalid payload length
  - Frame length is greater than MAX_FL
  - Frame received with a CRC-32 error
  - Frame truncated due to receive FIFO overflow
  - Frame is corrupted as PHY signaled an error (RX_ERR asserted during reception)
- An IP frame is detected and the IP header checksum is wrong
- An IP frame with a valid IP header and a valid IP header checksum is detected, the protocol is known but the protocol-specific checksum is wrong

If one of the errors occurs and the IP accelerator function is configured to discard frames (ENET*n*_RACC), the frame is automatically discarded. Statistics are maintained normally and are not affected by this discard function.

42.5.8.5 IPv4 fragments

When an IPv4 IP fragment frame is received, only the IP header is inspected and its checksum verified. 32-bit alignment operates the same way on fragments as it does on normal IP frames, as specified above.

The IP fragment frame payload is not inspected for any protocol headers. As such, a protocol header would only exist in the very first fragment. To assist in protocol-specific checksum verification, the one's-complement sum is calculated on the IP payload (all bytes following the IP header) and provided with the frame status word.

The frame fragment status field (RxBD[FRAG]) is set to indicate a fragment reception, and the one's-complement sum of the IP payload is available in RxBD[Payload checksum].

Note

After all fragments have been received and reassembled, the application software can take advantage of the payload checksum delivered with the frame's status word to calculate the protocol-specific checksum of the datagram.

For example, if a TCP payload is delivered by multiple IP fragments, the application software can calculate the pseudo-header checksum value from the first fragment, and add the payload checksums delivered with the status for all fragments to verify the TCP datagram checksum.

42.5.8.6 IPv6 support

The following sections describe the IPv6 support.

42.5.8.6.1 Receive processing

An Ethernet frame of type 0x86DD identifies an IP Version 6 frame (IPv6) frame. If an IPv6 frame is received, the first IP header is inspected (first ten words), which is available in every IPv6 frame.

If the receive SHIFT16 function is enabled, the IP header is aligned on a 32-bit boundary allowing more efficient processing (see 32-bit Ethernet payload alignment).

For TCP and UDP datagrams, the pseudo-header checksum calculation is performed and verified.

To assist in protocol-specific checksum verification, the one's-complement sum is always calculated on the IP payload (all bytes following the IP header) and provided with the frame status word. For example, if extension headers were present, their sums can be subtracted in software from the checksum to isolate the TCP/UDP datagram checksum, if required.

42.5.8.6.2 Transmit processing

For IPv6 transmission, the SHIFT16 function is supported to process 32-bit aligned datagrams.

IPv6 has no IP header checksum; therefore, the IP checksum insertion configuration is ignored.

The protocol checksum is inserted only if the next header of the IP header is a known protocol (TCP, UDP, or ICMP). If a known protocol is detected, the checksum over all bytes following the IP header is calculated and inserted in the correct position.

The pseudo-header checksum calculation is performed for TCP and UDP datagrams accordingly.

42.5.9 Resets and stop controls

The following sections describe the resets and stop controls.

42.5.9.1 Hardware reset

To reset the Ethernet module, set ENET*n*_ECR[RESET].

42.5.9.2 Soft reset

When ENET*n*_ECR[ETHER_EN] is cleared during operation, the following occurs:

- DMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers.
- A currently ongoing transmit is terminated by asserting TXER to the PHY.

- A currently ongoing transmit FIFO write from the application is terminated by stopping the write to the FIFO, and all further data from the application is ignored. All subsequent writes are ignored until re-enabled.
- A currently ongoing receive FIFO read is terminated. The RxBD has arbitrary values in this case.

42.5.9.3 Hardware freeze

When the processor enters debug mode and ECR[DBGEN] is set, the MAC enters a freeze state where it stops all transmit and receive activities gracefully.

The following happens when the MAC enters hardware freeze:

- A currently ongoing receive transaction on the receive application interface is completed as normal. No further frames are read from the FIFO.
- A currently ongoing transmit transaction on the transmit application interface is completed as normal (in other words, until writing end-of-packet (EOP)).
- A currently ongoing frame receive is completed normally, after which no further frames are accepted from the RMII.
- A currently ongoing frame transmit is completed normally, after which no further frames are transmitted.

42.5.9.4 Graceful stop

During a graceful stop, any currently ongoing transactions are completed normally and no further frames are accepted. The MAC can resume from a graceful stop without the need for a reset (for example, clearing ETHER_EN is not required).

The following conditions lead to a graceful stop of the MAC transmit or receive datapaths.

42.5.9.4.1 Graceful transmit stop (GTS)

When gracefully stopped, the MAC is no longer reading frame data from the transmit FIFO and has completed any ongoing transmission.

In any of the following conditions, the transmit datapath stops after an ongoing frame transmission has been completed normally.

- ENET*n*_TCR[GTS] is set by software.
- ENET*n*_TCR[TFC_PAUSE] is set by software requesting a pause frame transmission. The status (and register bit) is cleared after the pause frame has been sent.
- A pause frame was received stopping the transmitter. The stopped situation is terminated when the pause timer expires or a pause frame with zero quanta is received.
- MAC is placed in Sleep mode by software or the processor entering Stop mode (see Sleep mode).
- The MAC is in Hardware Freeze mode.

When the transmitter has reached its stopped state, the following events occur:

- The GRA interrupt is asserted, when transitioned into stopped.
- In Hardware Freeze mode, the GRA interrupt does not wait for the application write completion and asserts when the transmit state machine (in other words, line side of TX FIFO) reaches its stopped state.

42.5.9.4.2 Graceful receive stop (GRS)

When gracefully stopped, the MAC is no longer writing frames into the receive FIFO.

The receive datapath stops after any ongoing frame reception has been completed normally, if any of the following conditions occur:

- MAC is placed in Sleep mode either by the software or the processor is in Stop mode). The MAC continues to receive frames and search for magic packets if enabled (see Magic packet detection). However, no frames are written into the receive FIFO, and therefore are not forwarded to the application.
- The MAC is in Hardware Freeze mode. The MAC does not accept any frames from the RMII.

When the receive datapath is stopped, the following events occur:

- If the RX is in the stopped state, RCR[GRS] is set

- The GRA interrupt is asserted when the transmitter and receiver are stopped
- Any ongoing receive transaction to the application (RX FIFO read) continues normally until the frame end of package (EOP) is reached. After this, the following occurs:
  - When Sleep mode is active, all further frames are discarded, flushing the RX FIFO
  - In Hardware Freeze mode, no further frames are delivered to the application and they stay in the receive FIFO.

Note

The assertion of GRS does not wait for an ongoing FIFO read transaction on the application side of the FIFO (FIFO read).

42.5.9.4.3 Graceful stop interrupt (GRA)

The graceful stopped interrupt (GRA) is asserted for the following conditions:

- In Sleep mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- In Hardware Freeze mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- The MAC transmit datapath is stopped for any other condition (GTS, TFC_PAUSE, pause received).

The GRA interrupt is triggered only once when the stopped state is entered. If the interrupt is cleared while the stop condition persists, no further interrupt is triggered.

42.5.10 IEEE 1588 functions

To allow for IEEE 1588 or similar time synchronization protocol implementations, the MAC is combined with a time-stamping module to support precise time-stamping of incoming and outgoing frames. Set ENET*n*_ECR[1588EN] to enable 1588 support.

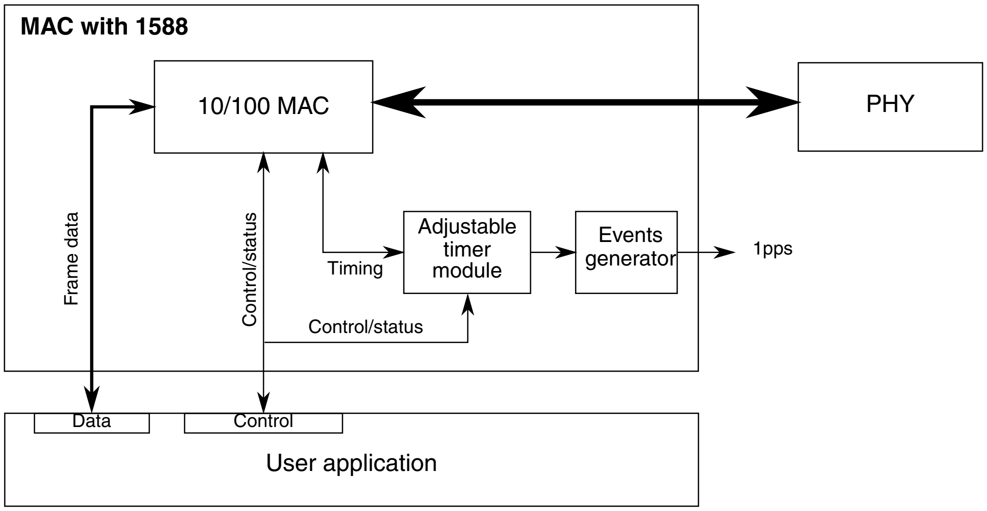


Figure 42-159. IEEE 1588 functions overview

42.5.10.1 Adjustable timer module

The adjustable timer module (TSM) implements the free-running counter (FRC), which generates the timestamps. The FRC operates with the time-stamping clock, which can be set to any value depending on your system requirements.

ENET time-stamp clock can be sourced either externally from ENET_1588_CLKIN PAD or from inner anadig ENET PLL. However, choose a period that is an integer value (e.g. 5 ns, 6 ns, 8 ns) to implement a precise timer.

Through dedicated correction logic, the timer can be adjusted to allow synchronization to a remote master and provide a synchronized timing reference to the local system. The timer can be configured to cause an interrupt after a fixed time period, to allow synchronization of software timers or perform other synchronized system functions.

The timer is typically used to implement a period of one second; hence, its value ranges from 0 to (1×10^9)-1. The period event can trigger an interrupt, and software can maintain the seconds and hours time values as necessary.

42.5.10.1.1 Adjustable timer implementation

The adjustable timer consists of a programmable counter/accumulator and a correction counter. The periods of both counters and their increment rates are freely configurable, allowing very fine tuning of the timer.

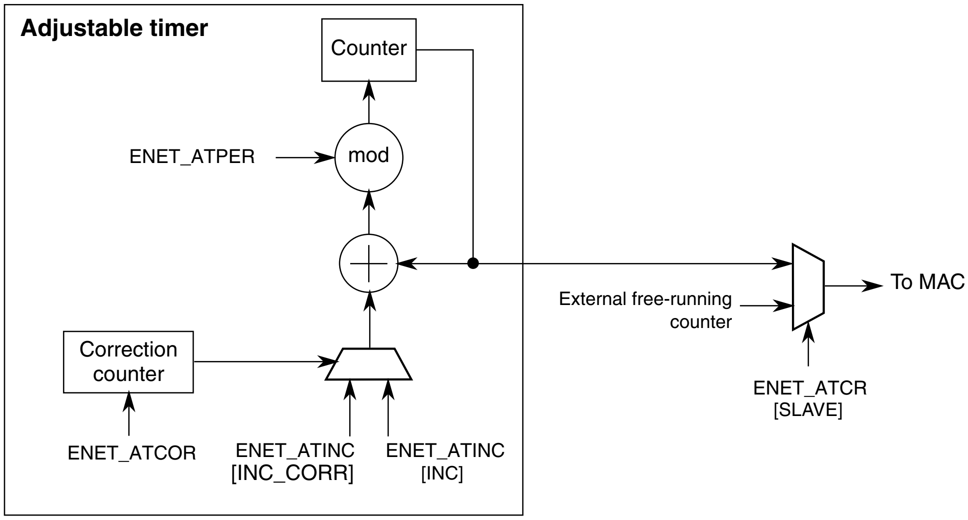


Figure 42-160. Adjustable timer implementation detail

The counter produces the current time. During each time-stamping clock cycle, a constant value is added to the current time as programmed in ENET*n*_ATINC. The value depends on the chosen time-stamping clock frequency. For example, if it operates at 125 MHz, setting the increment to eight represents 8 ns.

The period, configured in ENET*n*_ATPER, defines the modulo when the counter wraps. In a typical implementation, the period is set to 1×10^9 so that the counter wraps every second, and hence all timestamps represent the absolute nanoseconds within the one second period. When the period is reached, the counter wraps to start again respecting the period modulo. This means it does not necessarily start from zero, but instead the counter is loaded with the value (Current + Inc $-(1 \times 10^9)$), assuming the period is set to 1×10^9.

The correction counter operates fully independently, and increments by one with each time-stamping clock cycle. When it reaches the value configured in ENET*n*_ATCOR, it restarts and instructs the timer once to increment by the correction value, instead of the normal value.

The normal and correction increments are configured in ENET*n*_ATINC. To speed up the timer, set the correction increment more than the normal increment value. To slow down the timer, set the correction increment less than the normal increment value.

The correction counter only defines the distance of the corrective actions, not the amount. This allows very fine corrections and low jitter (in the range of 1 ns) independent of the chosen clock frequency.

By enabling slave mode (ENET*n*_ATCR[SLAVE] = 1), the timer is ignored and the current time is externally provided from one of the external modules. See the Chip Configuration details for which clock source is used. This is useful if multiple modules within the system must operate from a single timer. When slave mode is enabled, you still must set ENET*n*_ATINC[INC] to the value of the master, since it is used for internal comparisons.

42.5.10.2 Transmit timestamping

Only 1588 event frames need to be time-stamped on transmit. The client application (for example, the MAC driver) should detect 1588 event frames and set TxBD[TS] together with the frame.

If TxBD[TS] is set, the MAC records the timestamp for the frame in ENET*n*_ATSTMP. ENET*n*_EIR[TS_AVAIL] is set to indicate that a new timestamp is available.

Software implements a handshaking procedure by setting TxBD[TS] when it transmits the frame for which a timestamp is needed, and then waits for ENET*n*_EIR[TS_AVAIL] to determine when the timestamp is available. The timestamp is then read from ENET*n*_ATSTMP. This is done for all event frames. Other frames do not use TxBD[TS] and, therefore, do not interfere with the timestamp capture.

42.5.10.3 Receive timestamping

When a frame is received, the MAC latches the value of the timer when the frame's start of frame delimiter (SFD) field is detected, and provides the captured timestamp on RxBD[1588 timestamp]. This is done for all received frames.

42.5.10.4 Time synchronization

The adjustable timer module is available to synchronize the local clock of a node to a remote master. It implements a free running 32-bit counter, and also contains an additional correction counter.

The correction counter increases or decreases the rate of the free running counter, enabling very fine granular changes of the timer for synchronization, yet adding only very low jitter when performing corrections.

The application software implements, in a slave scenario, the required control algorithm, setting the correction to compensate for local oscillator drifts and locking the timer to the remote master clock on the network.

The timer and all timestamp-related information should be configured to show the true nanoseconds value of a second (in other words, the timer is configured to have a period of one second). Hence, the values range from 0 to $(1 \times 10^9)-1$. In this application, the seconds counter is implemented in software using an interrupt function that is executed when the nanoseconds counter wraps at 1×10^9.

42.5.11 FIFO thresholds

The core FIFO thresholds are fully programmable to dynamically change the FIFO operation.

For example, store and forward transfer can be enabled by a simple change in the FIFO threshold registers.

The thresholds are defined in 64-bit words.

42.5.11.1 Receive FIFO

Four programmable thresholds are available, which can be set to any value to control the core operation as follows.

Table 42-183. Receive FIFO thresholds definition

Register	Description
ENET*n*_RSFL [RX_SECTION_FULL]	When the FIFO level reaches the ENET*n*_RSFL value, the MAC status signal is asserted to indicate that data is available in the receive FIFO (cut-through operation). Once asserted, if the FIFO empties below the threshold set with ENET*n*_RAEM and if the end-of-frame is not yet stored in the FIFO, the status signal is deasserted again. If a frame has a size smaller than the threshold (in other words, an end-of-frame is available for the frame), the status is also asserted. To enable store and forward on the receive path, clear ENET*n*_RSFL. The MAC status signal is asserted only when a complete frame is stored in the receive FIFO. When programming a non-zero value to ENET*n*_RSFL (cut-through operation) it should be greater than ENET*n*_RAEM.
ENET*n*_RAEM [RX_ALMOST_EMPTY]	When the FIFO level reaches the ENET*n*_RAEM value, and the end-of-frame has not been received, the core receive read control stops the FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. Set ENET*n*_RAEM to a minimum of six.

Table continues on the next page...

Table 42-183. Receive FIFO thresholds definition (continued)

Register	Description
ENET*n*_RAFL [RX_ALMOST_FULL]	When the FIFO level approaches the maximum and there is no more space remaining for at least ENETn_RAFL number of words, the MAC control logic stops writing data in the FIFO and truncates the receive frame to avoid FIFO overflow. The corresponding error status is set when the frame is delivered to the application. Set ENET*n*_RAFL to a minimum of 4.
ENET*n*_RSEM [RX_SECTION_EMPTY]	When the FIFO level reaches the ENET*n*_RSEM value, an indication is sent to the MAC transmit logic, which generates an XOFF pause frame. This indicates FIFO congestion to the remote Ethernet client. When the FIFO level goes below the value programmed in ENET*n*_RSEM, an indication is sent to the MAC transmit logic, which generates an XON pause frame. This indicates the FIFO congestion is cleared to the remote Ethernet client. Clearing ENET*n*_RSEM disables any pause frame generation.

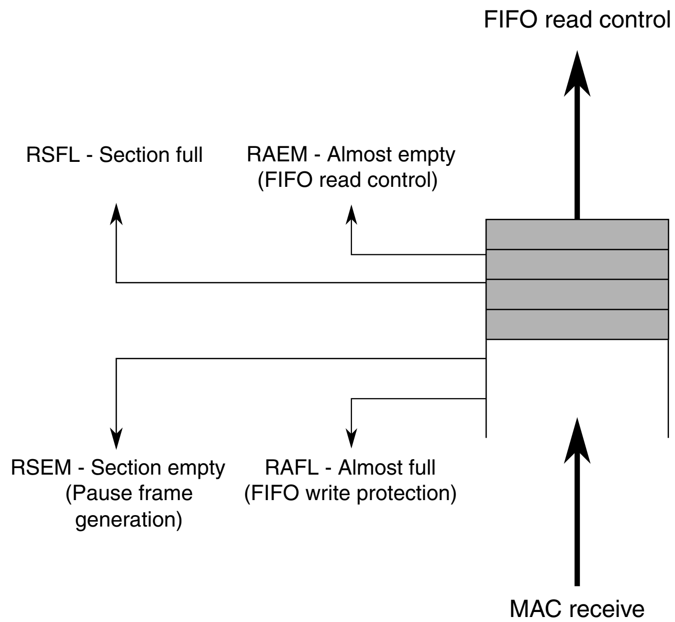


Figure 42-161. Receive FIFO overview

42.5.11.2 Transmit FIFO

Four programmable thresholds are available which control the core operation as described below.

Table 42-184. Transmit FIFO thresholds definition

Register	Description
ENET*n*_TAEM [TX_ALMOST _EMPTY]	When the FIFO level reaches the ENET*n*_TAEM value and no end-of-frame is available for the frame, the MAC transmit logic avoids a FIFO underflow by stopping FIFO reads and transmitting the Ethernet frame with an MII error indication. Set ENET*n*_TAEM to a minimum of 4.
ENET*n*_TAFL [TX_ALMOST _FULL]	When the FIFO level approaches the maximum, so that there is no more space for at least ENET*n*_TAFL number of words, the MAC deasserts its control signal to the application. If the application does not react on this signal, the FIFO write control logic avoids FIFO overflow by truncating the current frame and setting the error status. As a result, the frame is transmitted with an MII error indication. Set ENET*n*_TAFL to a minimum of 4. Larger values allow more latency for the application to react on the MAC control signal deassertion, before the frame is truncated. A typical setting is 8, which offers 3–4 clock cycles of latency to the application to react on the MAC control signal deassertion.
ENET*n*_TSEM [TX_SECTION _EMPTY]	When the FIFO level reaches the ENET*n*_TSEM value, a MAC status signal is deasserted to indicate that the transmit FIFO is getting full. This gives the application an indication to slow or stop its write transaction to avoid a buffer overflow. This is a pure indication function to the application. It has no effect within the MAC. When ENET*n*_TSEM is 0, the signal is never deasserted.
ENET*n*_TFWR	When the FIFO level reaches the ENET*n*_TFWR value and when STRFWD is cleared, the MAC transmit control logic starts frame transmission before the end-of-frame is available in the FIFO (cut-through operation). If a complete frame has a size smaller than the ENET*n*_TFWR threshold, the MAC also transmits the frame to the line. To enable store and forward on the transmit path, set STRFWD. In this case, the MAC starts to transmit data only when a complete frame is stored in the transmit FIFO.

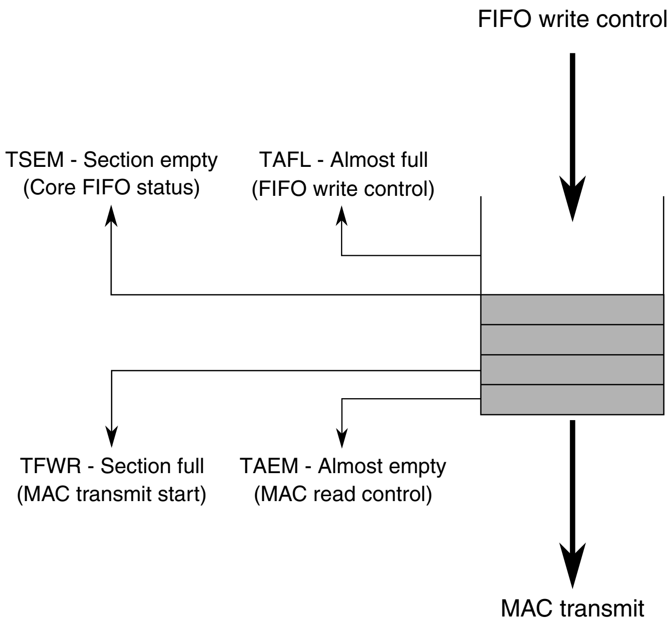


Figure 42-162. Transmit FIFO overview

42.5.12 Loopback options

The core implements external and internal loopback options, which are controlled by the ENET*n*_RCR register fields found here.

The core implements external and internal loopback options, which are controlled by the following ENET*n*_RCR register fields:

Table 42-185. Loopback options

Register field	Description
LOOP	Internal MII loopback. The MAC transmit is returned to the MAC receive. No data is transmitted to the external interfaces. In MII internal loopback, MII_TXCLK and MII_RXCLK must be provided with a clock signal (2.5 MHz for 10 Mbit/s, and 25 MHz for 100 Mbit/s))

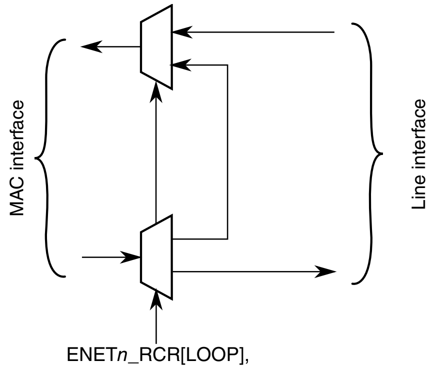


Figure 42-163. Loopback options

42.5.13 Legacy buffer descriptors

To support the Ethernet controller on previous Freescale devices, legacy FEC buffer descriptors are available. To enable legacy support, clear ENET*n*_ECR[1588EN].

NOTE

- Legacy buffer descriptors are used only in single-ring mode, that is, when DMA*n*CFG[DMA_CLASS_EN] are zero.
- The legacy buffer descriptor tables show the byte order for little-endian chips. DBSWP must be set to 1 after reset to enable little-endian mode.

42.5.13.1 Legacy receive buffer descriptor

The following table shows the legacy FEC receive buffer descriptor. Table 42-189 contains the descriptions for each field.

Table 42-186. Legacy FEC receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer — low halfword															
Offset + 6	Rx data buffer pointer — high halfword															

42.5.13.2 Legacy transmit buffer descriptor

The following table shows the legacy FEC transmit buffer descriptor. Table 42-191 contains the descriptions for each field.

Table 42-187. Legacy FEC transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data Length															
Offset + 2	R	TO1	W	TO2	L	TC	ABC[1]	—	—	—	—	—	—	—	—	—
Offset + 4	Tx Data Buffer Pointer — low halfword															
Offset + 6	Tx Data Buffer Pointer — high halfword															

1. This field is not supported by the uDMA.

42.5.14 Enhanced buffer descriptors

This section provides a description of the enhanced operation of the driver/DMA via the buffer descriptors.

It is followed by a detailed description of the receive and transmit descriptor fields. To enable the enhanced features, set ENET*n*_ECR[1588EN].

NOTE

The enhanced buffer descriptor tables show the byte order for little-endian chips. DBSWP must be set to 1 after reset to enable little-endian mode.

42.5.14.1 Enhanced receive buffer descriptor

The following table shows the enhanced uDMA receive buffer descriptor. Table 42-189 contains the descriptions for each field.

Table 42-188. Enhanced uDMA receive buffer descriptor (RxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 4	Rx data buffer pointer – low halfword															
Offset + 6	Rx data buffer pointer – high halfword															

Table continues on the next page...

Table 42-188. Enhanced uDMA receive buffer descriptor (RxBD) (continued)

Offset + 8	VPCP			—	—	—	—	—	—	—	ICE	PCR	—	VLAN	IPV6	FRAG
Offset + A	ME	—	—	—	—	PE	CE	UC	INT	—	—	—	—	—	—	—
Offset + C	Payload checksum															
Offset + E	Header length					—	—	—	Protocol type							
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															
Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 42-189. Receive buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length. Written by the MAC. Data length is the number of octets written by the MAC into this BD's data buffer if L is cleared (the value is equal to EMRBR), or the length of the frame including CRC if L is set. It is written by the MAC once as the BD is closed.
Offset + 2	15 E	Empty. Written by the MAC (= 0) and user (= 1). 0 The data buffer associated with this BD is filled with received data, or data reception has aborted due to an error condition. The status and length fields have been updated as required. 1 The data buffer associated with this BD is empty, or reception is currently in progress.
Offset + 2	14 RO1	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 2	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location. 1 The next buffer descriptor is found at the location defined in ENET*n*_RDSR.
Offset + 2	12 RO2	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 2	11 L	Last in frame. Written by the uDMA. 0 The buffer is not the last in a frame. 1 The buffer is the last in a frame.
Offset + 2	10–9	Reserved, must be cleared.

Table continues on the next page...

Table 42-189. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 2	8 M	Miss. Written by the MAC. This field is set by the MAC for frames accepted in promiscuous mode, but flagged as a miss by the internal address recognition. Therefore, while in promiscuous mode, you can use the this field to quickly determine whether the frame was destined to this station. This field is valid only if the L and PROM bits are set. 0 The frame was received because of an address recognition hit. 1 The frame was received because of promiscuous mode. The information needed for this field comes from the promiscuous_miss(ff_rx_err_stat[26]) sideband signal.
Offset + 2	7 BC	Set if the DA is broadcast (FFFF_FFFF_FFFF).
Offset + 2	6 MC	Set if the DA is multicast and not BC.
Offset + 2	5 LG	Receive frame length violation. Written by the MAC. A frame length greater than RCR[MAX_FL] was recognized. This field is valid only if the L field is set. The receive data is not altered in any way unless the length exceeds TRUNC_FL bytes.
Offset + 2	4 NO	Receive non-octet aligned frame. Written by the MAC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error or a PHY error occurred. This field is valid only if the L field is set. If this field is set, the CR field is not set.
Offset + 2	3	Reserved, must be cleared.
Offset + 2	2 CR	Receive CRC or frame error. Written by the MAC. This frame contains a PHY or CRC error and is an integral number of octets in length. This field is valid only if the L field is set.
Offset + 2	1 OV	Overrun. Written by the MAC. A receive FIFO overrun occurred during frame reception. If this field is set, the other status fields, M, LG, NO, CR, and CL, lose their normal meaning and are zero. This field is valid only if the L field is set.
Offset + 2	0 TR	Set if the receive frame is truncated (frame length >TRUNC_FL). If the TR field is set, the frame must be discarded and the other error fields must be ignored because they may be incorrect.
Offset + 4	15–0 low	Receive data buffer pointer, low halfword
0ffset + 6	15–0 high	Receive data buffer pointer, high halfword[1]
Offset + 8	15–13 VPCP	VLAN priority code point. This field is written by the uDMA to indicate the frame priority level. Valid values are from 0 (best effort) to 7 (highest). This value can be used to prioritize different classes of traffic (e.g., voice, video, data). This field is only valid if the L field is set.
Offset + 8	12–6	Reserved, must be cleared.
Offset + 8	5 ICE	IP header checksum error. This is an accelerator option. This field is written by the uDMA. Set when either a non-IP frame is received or the IP header checksum was invalid. An IP frame with less than 3 bytes of payload is considered to be an invalid IP frame. This field is only valid if the L field is set.
Offset + 8	4 PCR	Protocol checksum error. This is an accelerator option. This field is written by the uDMA. Set when the checksum of the protocol is invalid or an unknown protocol is found and checksumming could not be performed. This field is only valid if the L field is set.
Offset + 8	3	Reserved, must be cleared.

Table continues on the next page...

Table 42-189. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 8	2 VLAN	VLAN. This is an accelerator option. This field is written by the uDMA. It means that the frame has a VLAN tag. This field is valid only if the L field is set.
Offset + 8	1 IPV6	IPV6 Frame. This field is written by the uDMA. This field indicates that the frame has an IPv6 frame type. If this field is not set it means that an IPv4 or other protocol frame was received. This field is valid only if the L field is set.
Offset + 8	0 FRAG	IPv4 Fragment.This is an accelerator option.This field is written by the uDMA. It indicates that the frame is an IPv4 fragment frame. This field is only valid when the L field is set.
Offset + A	15 ME	MAC error. This field is written by the uDMA. This field means that the frame stored in the system memory was received with an error (typically, a receive FIFO overflow). This field is only valid when the L field is set.
Offset + A	14–11	Reserved, must be cleared.
Offset + A	10 PE	PHY Error. This field is written by the uDMA. Set to "1"when the frame was received with an Error character on the PHY interface. The frame is invalid. This field is valid only when the L field is set.
Offset + A	9 CE	Collision. This field is written by the uDMA. Set when the frame was received with a collision detected during reception. The frame is invalid and sent to the user application. This field is valid only when the L field is set.
Offset + A	8 UC	Unicast. This field is written by the uDMA, and means that the frame is unicast. This field is valid regardless of whether the L field is set.
Offset + A	7 INT	Generate RXB/RXF interrupt. This field is set by the user to indicate that the uDMA is to generate an interrupt on the *dma_int_rxb* / *dma_int_rxfevent*.
Offset + A	6–0	Reserved, must be cleared.
Offset + C	15–0 Payload checksum	Internet payload checksum. This is an accelerator option. It is the one's complement sum of the payload section of the IP frame. The sum is calculated over all data following the IP header until the end of the IP payload. This field is valid only when the L field is set.
Offset + E	15–11 Header length	Header length. This is an accelerator option. This field is written by the uDMA. This field is the sum of 32-bit words found within the IP and its following protocol headers. If an IP datagram with an unknown protocol is found, then the value is the length of the IP header. If no IP frame or an erroneous IP header is found, the value is 0. The following values are minimum values if no header options exist in the respective headers: • ICMP/IP: 6 (5 IP header, 1 ICMP header) • UDP/IP: 7 (5 IP header, 2 UDP header) • TCP/IP: 10 (5 IP header, 5 TCP header) This field is only valid if the L field is set.
Offset + E	10–8	Reserved, must be cleared.
Offset + E	7–0 Protocol type	Protocol type. This is an accelerator option. The 8-bit protocol field found within the IP header of the frame. It is valid only when ICE is cleared. This field is valid only when the L field is set.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.

Table continues on the next page...

Table 42-189. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 14 Offset + 16	15–0 1588 timestamp	This value is written by the uDMA. It is only valid if the L field is set.
Offset + 18 – Offset + 1E	15–0	Reserved, must be cleared.

1. The receive buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 16. The buffer must reside in memory external to the MAC. The Ethernet controller never modifies this value.

42.5.14.2 Enhanced transmit buffer descriptor

Table 42-190. Enhanced transmit buffer descriptor (TxBD)

	Byte 1								Byte 0							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	Data length															
Offset + 2	R	TO1	W	TO2	L	TC	—	—	—	—	—	—	—	—	—	—
Offset + 4	Tx Data Buffer Pointer – low halfword															
Offset + 6	Tx Data Buffer Pointer – high halfword															
Offset + 8	TXE	—	UE	EE	FE	LCE	OE	TSE	—	—	—	—	—	—	—	—
Offset + A	—	INT	TS	PINS	IINS	—	—	—	—	—	—	—	—	—	—	—
Offset + C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – low halfword															
Offset + 16	1588 timestamp – high halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 42-191. Enhanced transmit buffer descriptor field definitions

Word	Field	Description
Offset + 0	15–0 Data Length	Data length, written by user. Data length is the number of octets the MAC should transmit from this BD's data buffer. It is never modified by the MAC.

Table continues on the next page...

Table 42-191. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 2	15 R	Ready. Written by the MAC and you. 0 — The data buffer associated with this BD is not ready for transmission. You are free to manipulate this BD or its associated data buffer. The MAC clears this field after the buffer has been transmitted or after an error condition is encountered. 1 — The data buffer, prepared for transmission by you, has not been transmitted or currently transmits. You may write no fields of this BD after this field is set.
Offset + 2	14 TO1	Transmit software ownership. This field is reserved for software use. This read/ write field is not modified by hardware and its value does not affect hardware.
Offset + 2	13 W	Wrap. Written by user. 0 — The next buffer descriptor is found in the consecutive location 1 — The next buffer descriptor is found at the location defined in ETDSR.
Offset + 2	12 TO2	Transmit software ownership. This field is reserved for use by software. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 2	11 L	Last in frame. Written by user. 0 — The buffer is not the last in the transmit frame 1 — The buffer is the last in the transmit frame
Offset + 2	10 TC	Transmit CRC. Written by user, and valid only when L is set. 0 — End transmission immediately after the last data byte 1 — Transmit the CRC sequence after the last data byte This field is valid only when the L field is set.
Offset + 2	9 ABC	Append bad CRC. **Note:** This field is not supported by the uDMA and is ignored.
Offset + 2	8–0	Reserved, must be cleared.
Offset + 4	15–0 low	Tx data buffer pointer, low halfword
Offset + 6	15–0 high	Tx data buffer pointer, high halfword. The transmit buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 8. The buffer must reside in memory external to the MAC. This value is never modified by the Ethernet controller.
Offset + 8	15 TXE	Transmit error occurred. This field is written by the uDMA. This field indicates that there was a transmit error of some sort reported with the frame. Effectively this field is an OR of the other error fields including UE, EE, FE, LCE, OE, and TSE. This field is valid only when the L field is set.
Offset + 8	14	Reserved, must be cleared.
Offset + 8	13 UE	Underflow error. This field is written by the uDMA. This field indicates that the MAC reported an underflow error on transmit. This field is valid only when the L field is set.
Offset + 8	12 EE	Excess Collision error. This field is written by the uDMA. This field indicates that the MAC reported an excess collision error on transmit. This field is valid only when the L field is set.

Table continues on the next page...

Table 42-191. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 8	11 FE	Frame with error. This field is written by the uDMA. This field indicates that the MAC reported that the uDMA reported an error when providing the packet. This field is valid only when the L field is set.
Offset + 8	10 LCE	Late collision error. This field is written by the uDMA. This field indicates that the MAC reported that there was a Late Collision on transmit. This field is valid only when the L field is set.
Offset + 8	9 OE	Overflow error. This field is written by the uDMA. This field indicates that the MAC reported that there was a FIFO overflow condition on transmit. This field is only valid when the L field is set.
Offset + 8	8 TSE	Timestamp error. This field is written by the uDMA. This field indicates that the MAC reported a different frame type then a timestamp frame. This field is valid only when the L field is set.
Offset + 8	7–0	Reserved, must be cleared.
Offset + A	15	Reserved, must be cleared.
Offset + A	14 INT	Generate interrupt flags. This field is written by the user. This field is valid regardless of the L field and must be the same for all EBD for a given frame. The uDMA does not update this value.
Offset + A	13 TS	Timestamp. This field is written by the user. This indicates that the uDMA is to generate a timestamp frame to the MAC. This field is valid regardless of the L field and must be the same for all EBD for the given frame. The uDMA does not update this value.
Offset + A	12 PINS	Insert protocol specific checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the protocol checksum and overwrites the corresponding checksum field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + A	11 IINS	Insert IP header checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the IP header checksum and overwrites the corresponding header field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + A	10–0	Reserved, must be cleared.
Offset + C	15–0	Reserved, must be cleared.
Offset + E	15–0	Reserved, must be cleared.
Offset + 10	15–0	Reserved, must be cleared.
Offset + 12	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 12	14–0	Reserved, must be cleared.
Offset + 14 Offset + 16	15–0 1588 timestamp	This value is written by the uDMA . It is valid only when the L field is set.
Offset + 18–Offset + 1E	15–0	Reserved, must be cleared.

42.5.15 Client FIFO application interface

The FIFO interface is completely asynchronous from the Ethernet line, and the transmit and receive interface can operate at a different clock rate.

All transfers to/from the user application are handled independently of the core operation, and the core provides a simple interface to user applications based on a two-signal handshake.

42.5.15.1 Data structure description

The data structure defined in the following tables for the FIFO interface must be respected to ensure proper data transmission on the Ethernet line. Byte 0 is sent to and received from the line first.

Table 42-192. FIFO interface data structure

	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Word 0	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Word 1	Byte 15	Byte 14	Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8
...	...							

The size of a frame on the FIFO interface may not be a modulo of 64-bit.

The user application may not care about the Ethernet frame formats in full detail. It needs to provide and receive an Ethernet frame with the following structure:

- Ethernet MAC destination address
- Ethernet MAC source address
- Optional 802.1q VLAN tag (VLAN type and info field)
- Ethernet length/type field
- Payload

Frames on the FIFO interface do not contain preamble and SFD fields, which are inserted and discarded by the MAC on transmit and receive, respectively.

- On receive, CRC and frame padding can be stripped or passed through transparently.
- On transmit, padding and CRC can be provided by the user application, or appended automatically by the MAC independently for each frame. No size restrictions apply.

Note

On transmit, if ENET*n*_TCR[ADDINS] is set, bytes 6–11 of each frame can be set to any value, since the MAC overwrites the bytes with the MAC address programmed in the ENET*n*_PAUR and ENET*n*_PALR registers.

Table 42-193. FIFO interface frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–13	Length/type field
14–N	Payload data

VLAN-tagged frames are supported on both transmit and receive, and implement additional information (VLAN type and info).

Table 42-194. FIFO interface VLAN frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–15	VLAN tag and info
16–17	Length/type field
18–N	Payload data

Note

The standard defines that the LSB of the MAC address is sent/received first, while for all the other header fields — in other words, length/type, VLAN tag, VLAN info, and pause quanta — the MSB is sent/received first.

42.5.15.2 Data structure examples

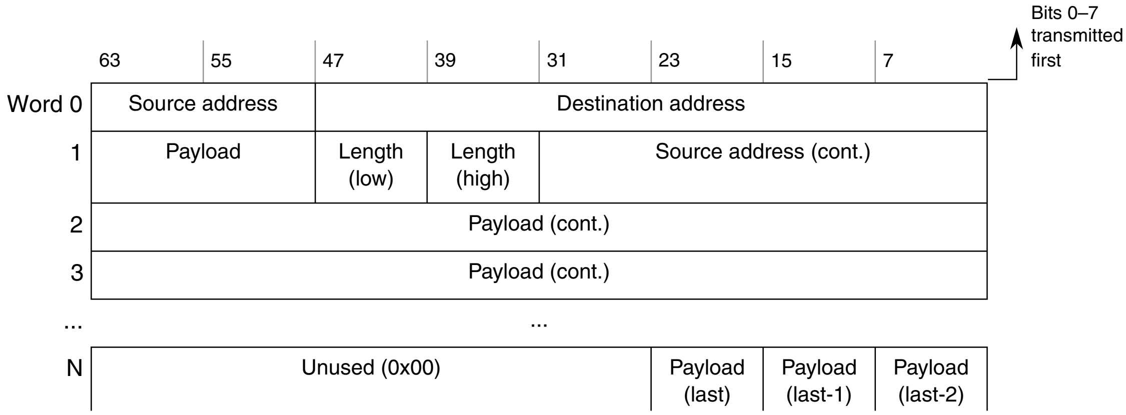


Figure 42-164. Normal Ethernet frame 64-bit mapping example

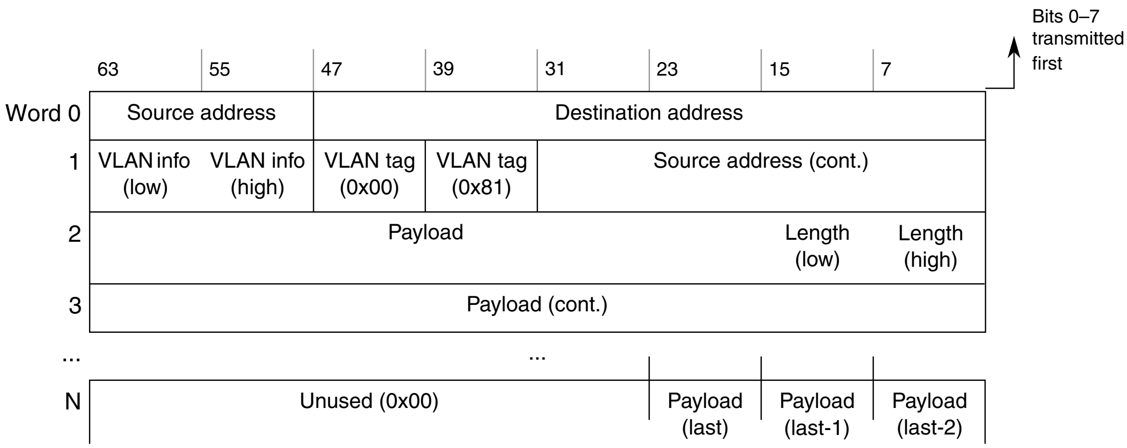


Figure 42-165. VLAN-tagged frame 64-bit mapping example

If CRC forwarding is enabled (CRCFWD = 0), the last four valid octets of the frame contain the FCS field. The non-significant bytes of the last word can have any value.

42.5.15.3 Frame status

A MAC layer status word and an accelerator status word is available in the receive buffer descriptor.

See Enhanced buffer descriptors for details.

The status is available with each frame with the last data of the frame.

If the frame status contains a MAC layer error (for example, CRC or length error), RxBD[ME] is also set with the last data of the frame.

42.5.16 FIFO protection

The following sections describe the FIFO protection mechanisms.

42.5.16.1 Transmit FIFO underflow

During a frame transfer, when the transmit FIFO reaches the almost empty threshold with no end-of-frame indication stored in the FIFO, the MAC logic:

- Stops reading data from the FIFO
- Asserts the MII error signal (MII_TXER) (1 in Figure 42-166) to indicate that the fragment already transferred is not valid
- Deasserts the MII transmit enable signal (MII_TXEN) to terminate the frame transfer (2)

After an underflow, when the application completes the frame transfer (3), the MAC transmit logic discards any new data available in the FIFO until the end of packet is reached (4) and sets the enhanced TxBD[UE] field.

The MAC starts to transfer data on the MII interface when the application sends a new frame with a start of frame indication (5).

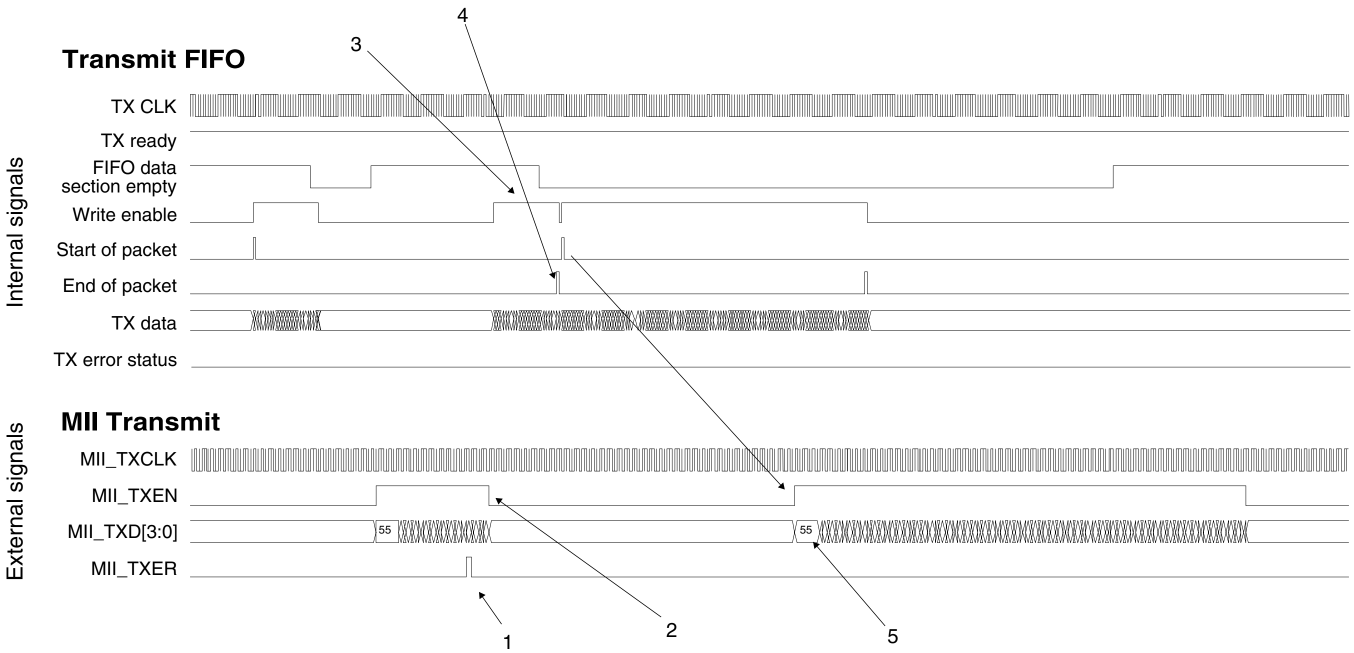


Figure 42-166. Transmit FIFO underflow protection

42.5.16.2 Transmit FIFO overflow

On the transmit path, when the FIFO reaches the programmable almost full threshold, the internal MAC ready signal is deasserted. The application should stop sending new data.

However, if the application keeps sending data, the transmit FIFO overflows, corrupting contents that were previously stored. The core logic sets the enhanced TxBD[OE] field for the next frame transmitted to indicate this overflow occurence.

Note

Overflow is a fatal error and must be addressed by resetting the core or clearing ENET*n*_ECR[ETHER_EN], to clear the FIFOs and prepare for normal operation again.

42.5.16.3 Receive FIFO overflow

During a frame reception, if the client application is not able to receive data (1), the MAC receive control truncates the incoming frame when the FIFO reaches the programmable almost-full threshold to avoid an overflow.

The frame is subsequently received on the FIFO interface with an error indication (enhanced RxBD[ME] field set together with receive end-of-packet) (2) with the truncation error status field set (3).

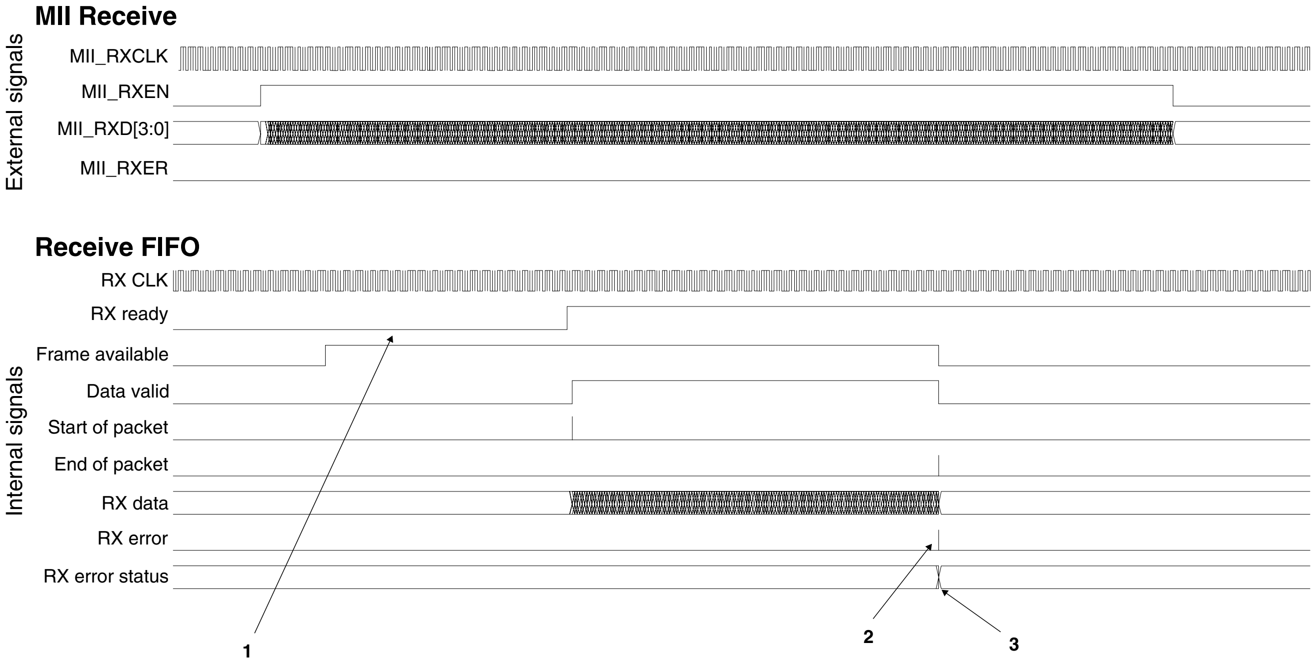


Figure 42-167. Receive FIFO overflow protection

42.5.17 Reference clock

The input clocks to the ENET module must meet the specifications in the following table.

Ethernet speed mode	Ethernet bus clock	Minimum ENET system clock
10 Mbps	2.5 MHz	5 MHz
100 Mbps	25.0 MHz	50 MHz

42.5.18 PHY management interface

The MDIO interface is a two-wire management interface. The MDIO management interface implements a standardized method to access the PHY device management registers.

The core implements a master MDIO interface, which can be connected to up to 32 PHY devices.

42.5.18.1 MDIO frame format

The core MDIO master controller communicates with the slave (PHY device) using frames that are defined in the following table.

A complete frame has a length of 64 bits made up of an optional 32-bit preamble, 14-bit command, 2-bit bus direction change, and 16-bit data. Each bit is transferred on the rising edge of the MDIO clock (MDC signal).

The core PHY management interface supports the standard MDIO specification (IEEE803.2 Clause 22).

Table 42-195. MDIO frame formats (read/write)

Type	Command					TA	Data		Idle
	PRE	ST	OP	Addr1	Addr2		MSB	LSB	
Read	1…1	01	10	xxxxx	xxxxx	Z0	xxxxxxxxxxxxxxxx		Z
Write	1…1	01	01	xxxxx	xxxxx	10	xxxxxxxxxxxxxxxx		Z

Table 42-196. MDIO frame field descriptions

Field	Description
PRE	Preamble. 32 bits of logical ones sent prior to every transaction when ENET*n*_MSCR[DIS_PRE] is cleared. If DIS_PRE is set, the preamble is not generated.
ST	Start indication, programmed with ENET*n*_MMFR[ST] • Standard MDIO (Clause 22): 01
OP	Opcode defines if a read or write operation is performed, programmed with ENET*n*_MMFR[OP]. 01 Write operation 10 Read operation
Addr1	The PHY device address, programmed with ENET*n*_MMFR[PA]. Up to 32 devices can be addressed.
Addr2	Register address, programmed with ENET*n*_MMFR[RA]. Each PHY can implement up to 32 registers.
TA	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
Data	16 bits of data, set to ENET*n*_MMFR[DATA]. Written to or read from the PHY
Idle	The MDIO data signal is tri-stated between frames.

42.5.18.2 MDIO clock generation

The MDC clock is generated from the internal bus clock divided by the value programmed in ENET*n*_MSCR[MII_SPEED].

42.5.18.3 MDIO operation

To perform an MDIO access, set the MDIO command register (ENET*n*_MMFR) according to the description provided in MII Management Frame Register (ENETn_MMFR).

To check when the programmed access completes, read the ENET*n*_EIR[MII] field.

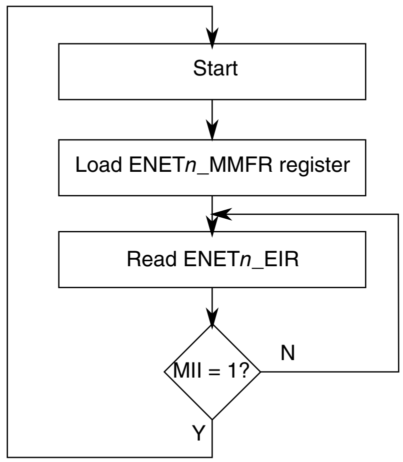


Figure 42-168. MDIO access overview

42.5.19 Ethernet interfaces

RMII 10/100 is implemented using interface converters/gaskets.

The following table shows how to configure ENET registers to select each interface.

Mode	ECR[SPEED]	RCR[RMII_10T]	RCR[RMII_MODE]
RMII - 10 Mbit/s	0	1	1
RMII - 100 Mbit/s	0	0	1

42.5.19.1 RMII interface

In RMII receive mode, for normal reception following assertion of CRS_DV, RXD[1:0] is 00 until the receiver determines that the receive event has a proper start-of-stream delimiter (SSD).

The preamble appears (RXD[1:0]=01) and the MACs begin capturing data following detection of SFD.

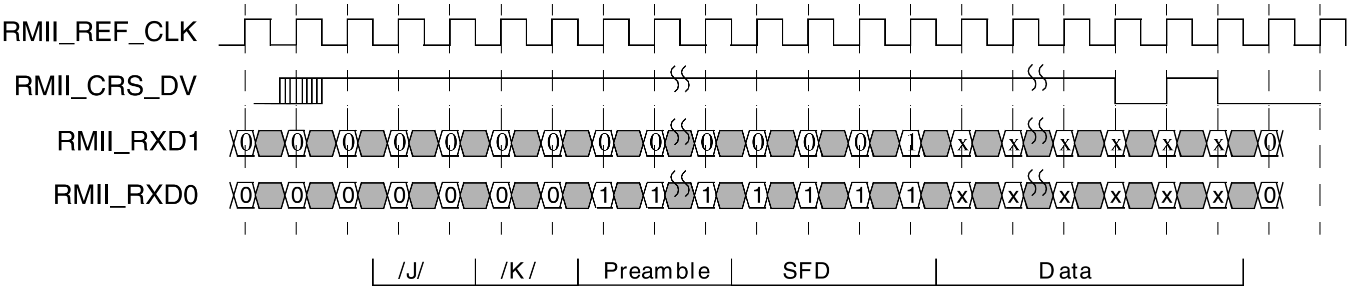


Figure 42-169. RMII receive operation

If a false carrier is detected (bad SSD), then RXD[1:0] is 10 until the end of the receive event. This is a unique pattern since a false carrier can only occur at the beginning of a packet where the preamble is decoded (RXD[1:0] = 01).

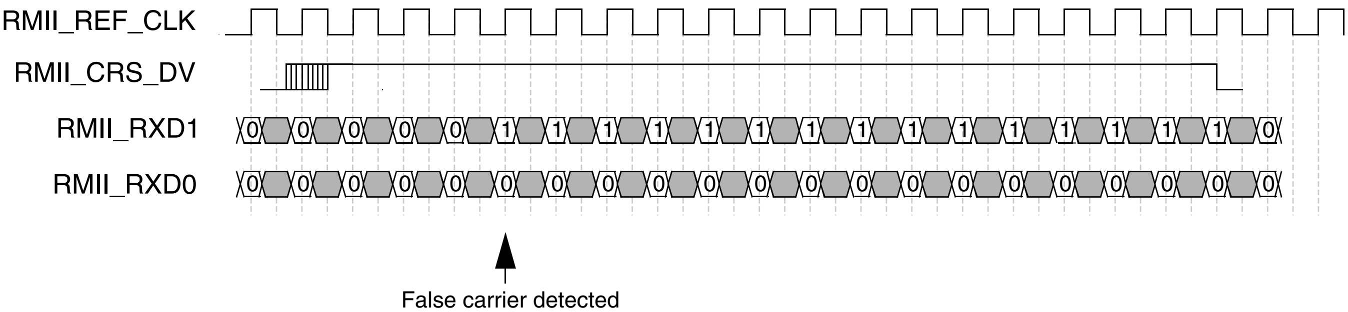


Figure 42-170. RMII receive operation with false carrier

In RMII transmit mode, TXD[1:0] provides valid data for each REF_CLK period while TXEN is asserted.

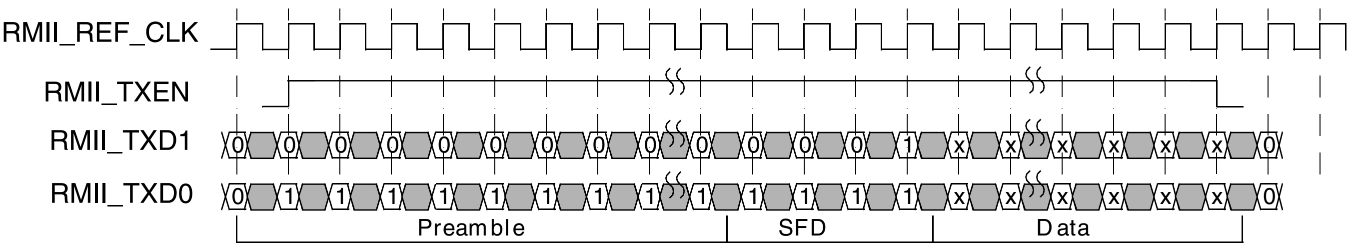
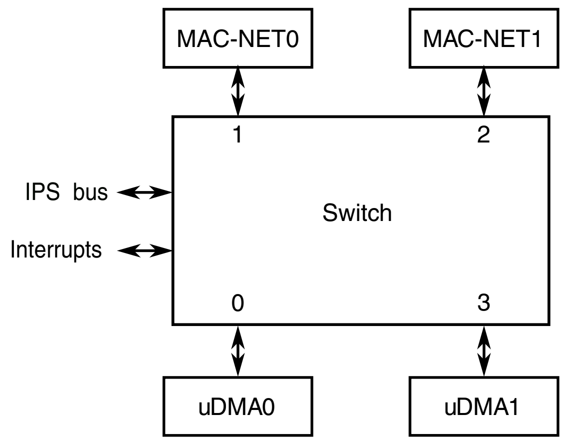


Figure 42-171. RMII transmit operation

Chapter 43
Ethernet Switch (ESW)

43.1 Introduction

The switch core is seamlessly connected to the MAC-NET core and DMA controllers. For control and configuration, the switch implements a register interface and multiple maskable interrupts.



The switch port assignment is listed in Table 43-1 .

Table 43-1. Port Assignment

Switch Port	Assignment
0	DMA 0
1	MAC-NET 0
2	MAC-NET 1
3 (bypass port)	DMA 1

43.1.1 Block Diagram

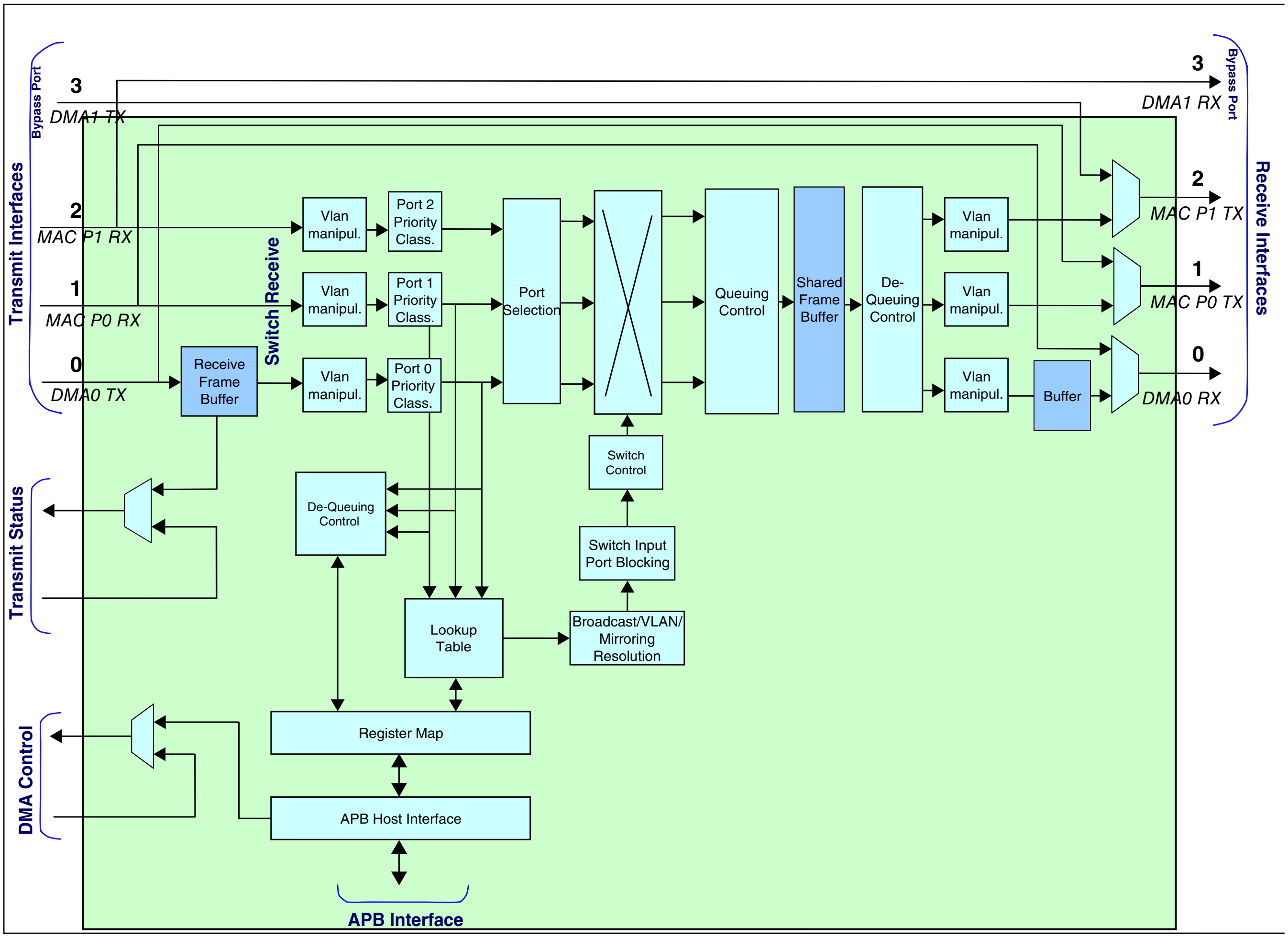


Figure 43-1. Block Diagram

Note

- The left side is termed "Transmit Interfaces" as the interfaces are driven by the DMA transmit interfaces in pass-through mode. If the switch is enabled, these interfaces connect to the switch internal receive interfaces. The right side is named "Receive Interfaces" as the interfaces represent (are driven from) the respective MAC receive interfaces in pass-through mode connected to the DMA RX. If the switch is enabled these are driven by the switch internal transmit interfaces. See Introduction for a description of the operational modes.
- All descriptions related to switch functions refer to the switch internal receive/transmit interfaces.

- Receive Frame buffer (DMA0 TX interface). Can hold at least one complete frame transferred from DMA0 to the switch. This is a local 64-bit wide FIFO to absorb a burst from the DMA, provide the necessary transaction handshake to the DMA, and forward the frame then to the switch logic.
- Decoupling buffer only (DMA0 RX interface) to absorb switch latency (e.g. 16/32 words) resulting from a rdy deassertion from DMA0.
- The APB host interface module implements an indirect addressing scheme to the internal registers to allow arbitrary length access cycles (e.g. address table read/write and configuration register read/write). However the DMA control registers are directly mapped and accessed through APB.

43.1.2 Features

- Integrated Ethernet switch engine compatible with 10/100 MAC-NET core
- Three port switch with a fourth DMA bypass port
- Can be configured to operate as a 3-port switch (switch mode) or as two independent ports (passthrough mode)
- Filters and forward traffic at wire-speed on all ports
- Per-queue tail-drop congestion management
- Implements hardware switching look-up mechanism providing a learning capacity of up to 2K MAC addresses
- Supports configurable VLAN switching when MAC address lookup should be omitted
- Classification and priority assignment based on port number, MAC address, IPv4 DiffServ code point field, IPv6 Class of Service and VLAN Priority (IEEE802.1q)
- Efficient output queue frame buffering with shared Frame buffer of 24 Kbyte
- Each port implements four priority queues with configurable weighted round-robin selection

- Support Ethernet multicast and broadcast with flooding control to avoid unnecessary duplication of frames
- Programmable multicast destination port mask to restrict frame duplication for individual multicast addresses
- Multicast and broadcast resolution with VLAN domain filtering providing a strict separation of up to 32 VLANs
- IP snooping with programmable protocol and port number registers
- Programmable ingress and egress VLAN tag addition, removal and manipulation supporting single and double-tagged VLAN frames
- Event and status signals which can monitor port activity, severe error conditions, or any user-specific event
- Programmable firmware operation with static or dynamic (learning, aging) switching tables
- Switch firmware source available to provide customer-specific software development capability
- Support for IEEE 1588 precise time-stamping applications
- Supports aggregation and redundant backplane applications

43.2 Modes of Operation

The switch can be configured to operate in the following modes using ESW_MODE[SWEN].

- Passthrough mode — The switch logic is disabled and bypassed
- Switch mode — The switch logic is enabled

43.2.1 Passthrough mode

When ESW_MODE[SWEN] is written to 0, the switch logic is bypassed and can be powered down and disabled with the switch clocks stopped and the switch reset signals asserted.

The switch APB interface and interrupt signals are disabled and should not be used. To control the frame transfer from DMA0 and DMA1, the MAC-NET 0 and the MAC-NET 1 APB interfaces and interrupt signals should be used.

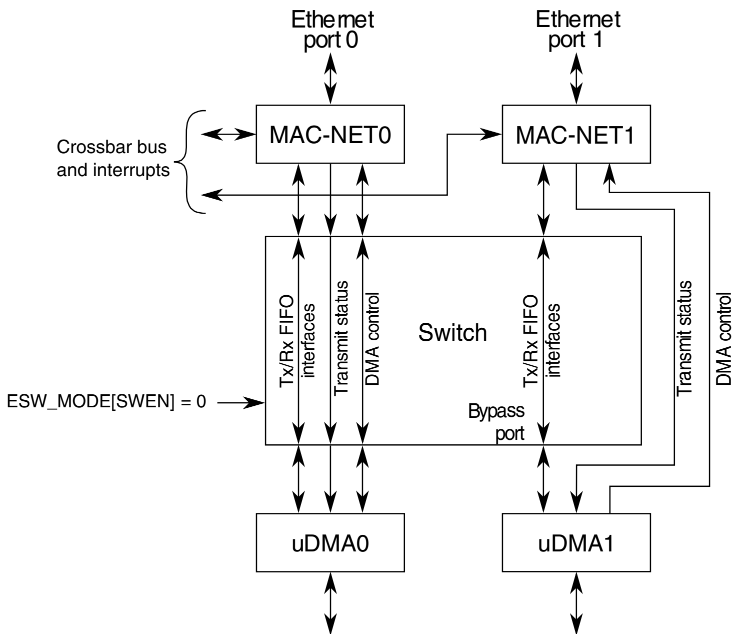


Figure 43-2. Passthrough Mode Configuration Overview

43.2.2 Switch Mode

When the switch is programmed to operate in switch mode (ESW_MODE[SWEN] set to 1), the bypass mode (port 1) interface is disabled and should not be used.

Frame transfers to and from the line are performed on port 0 only (DMA 0). The transmit status signals are generated from the switch port 0 receive buffer and the DMA control signals from the switch register space. The MAC-NET 0 and MAC-NET 1 transmit status and DMA control signals are not used.

The MAC-NET 0 and MAC-NET 1 APB interfaces and interrupts are enabled and can monitor the line activity and gather the line statistic information.

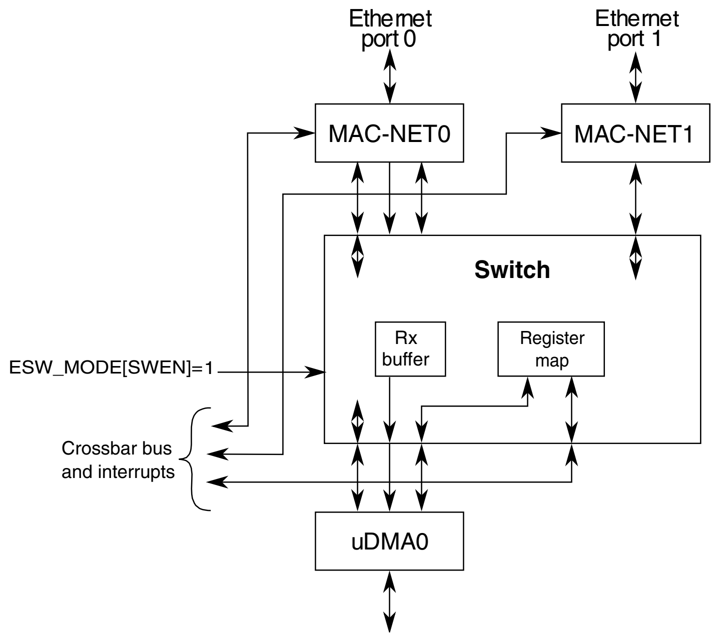


Figure 43-3. Switch Mode Configuration Overview

43.2.2.1 Port 0 Input Buffer

A dedicated input buffer of at least 2 KB storage is implemented at the port 0 (DMA 0) input interface, when the switch is operating in switch mode. In bypass mode, this buffer is bypassed.

The input buffer is normally operated in store-and-forward mode, absorbing a DMA burst and forwarding the frame to the switch (internally) when the complete frame is stored in the input buffer.

43.2.2.2 Port 0 Input Backpressure/Congestion Indication

When frames are transferred from DMA0 to the switch's port 0 transmit interface, the input buffer indicates when data is to be written to the port 0 input. When the buffer reaches an almost full threshold, it indicates a stop request to DMA0. DMA0 may write a few more words (typically up to four), and then stops writing.

In addition, a special backpressure mechanism for port 0 is included to pause DMA0 transfers when the output queues' shared memory (see Output Frame Queuing) becomes full to a programmable threshold. Respecting the output queues' shared memory fill level can avoid the switch (due to memory congestion) discarding frames written by the DMA0. The threshold is configured through ESW_P0BCT.

If the shared memory has less than ESW_P0BCT number of free cells available, the switch stops serving port 0. That is, the switch does not start to read a frame from the port 0 input buffer. The port 0 input buffer continues to accept data from DMA0 until it becomes full.

Note

The backpressure only considers the total amount of memory available, not a specific queue. Hence, it may still happen that an outgoing frame from DMA0 is discarded by the switch, if the output queue for the frame is congested while the total amount of memory has free space available.

The backpressure threshold (ESW_P0BCT) should be set higher than the memory full threshold (ESW_LMT) to stop the DMA0 before a memory full situation.

43.3 ESW memory map and registers

ESW memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400E_8000	Revision (ESW_REV)	32	R	See section	43.3.1/2014
400E_8004	Scratch register (ESW_SCR)	32	R/W	0000_0000h	43.3.2/2015
400E_8008	Port enable register (ESW_PER)	32	R/W	0000_0000h	43.3.3/2015
400E_8010	VLAN verify (ESW_VLANV)	32	R/W	0000_0000h	43.3.4/2016
400E_8014	Default broadcast resolution (ESW_DBCR)	32	R/W	0000_0000h	43.3.5/2018
400E_8018	Default multicast resolution (ESW_DMCR)	32	R/W	0000_0000h	43.3.6/2019
400E_801C	Blocking and learning enable (ESW_BKLR)	32	R/W	0000_0000h	43.3.7/2019
400E_8020	Bridge management port configuration (ESW_BMPC)	32	R/W	0000_0000h	43.3.8/2021
400E_8024	Mode configuration (ESW_MODE)	32	R/W	0000_0000h	43.3.9/2022
400E_8028	VLAN input manipulation select (ESW_VIMSEL)	32	R/W	0000_0000h	43.3.10/ 2023
400E_802C	VLAN output manipulation select (ESW_VOMSEL)	32	R/W	0000_0000h	43.3.11/ 2024
400E_8030	VLAN input manipulation enable (ESW_VIMEN)	32	R/W	0000_0000h	43.3.12/ 2025
400E_8034	VLAN tag ID (ESW_VID)	32	R/W	0000_8100h	43.3.13/ 2025
400E_8040	Mirror control register (ESW_MCR)	32	R/W	0000_0000h	43.3.14/ 2026
400E_8044	Egress port definitions (ESW_EGMAP)	32	R/W	0000_0000h	43.3.15/ 2027

Table continues on the next page...

ESW memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400E_8048	Ingress port definitions (ESW_INGMAP)	32	R/W	0000_0000h	43.3.16/ 2028
400E_804C	Ingress source MAC address low (ESW_INGSAL)	32	R/W	0000_0000h	43.3.17/ 2028
400E_8050	Ingress source MAC address high (ESW_INGSAH)	32	R/W	0000_0000h	43.3.18/ 2029
400E_8054	Ingress destination MAC address low (ESW_INGDAL)	32	R/W	0000_0000h	43.3.19/ 2029
400E_8058	Ingress destination MAC address high (ESW_INGDAH)	32	R/W	0000_0000h	43.3.20/ 2029
400E_805C	Egress source MAC address low (ESW_EGSAL)	32	R/W	0000_0000h	43.3.21/ 2030
400E_8060	Egress source MAC address high (ESW_EGSAH)	32	R/W	0000_0000h	43.3.22/ 2030
400E_8064	Egress destination MAC address low (ESW_EGDAL)	32	R/W	0000_0000h	43.3.23/ 2030
400E_8068	Egress destination MAC address high (ESW_EGDAH)	32	R/W	0000_0000h	43.3.24/ 2031
400E_806C	Mirror count value (ESW_MCVAL)	32	R/W	0000_0000h	43.3.25/ 2031
400E_8080	Memory manager status (ESW_MMSR)	32	R/W	0060_000Ah	43.3.26/ 2032
400E_8084	Low memory threshold (ESW_LMT)	32	R/W	0000_0009h	43.3.27/ 2033
400E_8088	Lowest number of free cells (ESW_LFC)	32	R/W	0000_0000h	43.3.28/ 2034
400E_808C	Port congestion status (ESW_PCSR)	32	R	0000_0000h	43.3.29/ 2034
400E_8090	Switch input and output interface status (ESW_IOSR)	32	R	0000_0000h	43.3.30/ 2035
400E_8094	Queue weights (ESW_QWT)	32	R/W	0000_0000h	43.3.31/ 2036
400E_809C	Port 0 Backpressure Congestion Threshold (ESW_P0BCT)	32	R/W	0000_0009h	43.3.32/ 2037
400E_80BC	Port 0 forced forwarding enable (ESW_FFEN)	32	R/W	0000_0000h	43.3.33/ 2038
400E_80C0	Port snooping registers (ESW_PSNP1)	32	R/W	0000_0000h	43.3.34/ 2039
400E_80C4	Port snooping registers (ESW_PSNP2)	32	R/W	0000_0000h	43.3.35/ 2040
400E_80C8	Port snooping registers (ESW_PSNP3)	32	R/W	0000_0000h	43.3.36/ 2041
400E_80CC	Port snooping registers (ESW_PSNP4)	32	R/W	0000_0000h	43.3.37/ 2042

Table continues on the next page...

ESW memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400E_80D0	Port snooping registers (ESW_PSNP5)	32	R/W	0000_0000h	43.3.38/ 2043
400E_80D4	Port snooping registers (ESW_PSNP6)	32	R/W	0000_0000h	43.3.39/ 2044
400E_80D8	Port snooping registers (ESW_PSNP7)	32	R/W	0000_0000h	43.3.40/ 2045
400E_80DC	Port snooping registers (ESW_PSNP8)	32	R/W	0000_0000h	43.3.41/ 2046
400E_80E0	IP snooping registers (ESW_IPSNP1)	32	R/W	0000_0000h	43.3.42/ 2047
400E_80E4	IP snooping registers (ESW_IPSNP2)	32	R/W	0000_0000h	43.3.43/ 2048
400E_80E8	IP snooping registers (ESW_IPSNP3)	32	R/W	0000_0000h	43.3.44/ 2049
400E_80EC	IP snooping registers (ESW_IPSNP4)	32	R/W	0000_0000h	43.3.45/ 2050
400E_80F0	IP snooping registers (ESW_IPSNP5)	32	R/W	0000_0000h	43.3.46/ 2051
400E_80F4	IP snooping registers (ESW_IPSNP6)	32	R/W	0000_0000h	43.3.47/ 2052
400E_80F8	IP snooping registers (ESW_IPSNP7)	32	R/W	0000_0000h	43.3.48/ 2053
400E_80FC	IP snooping registers (ESW_IPSNP8)	32	R/W	0000_0000h	43.3.49/ 2054
400E_8100	Port 0 VLAN priority resolution map (ESW_P0VRES)	32	R/W	0000_0000h	43.3.50/ 2055
400E_8104	Port 1 VLAN priority resolution map (ESW_P1VRES)	32	R/W	0000_0000h	43.3.51/ 2056
400E_8108	Port 2 VLAN priority resolution map (ESW_P2VRES)	32	R/W	0000_0000h	43.3.52/ 2057
400E_8140	IPv4/v6 priority resolution table (ESW_IPRES)	32	R/W	0000_0000h	43.3.53/ 2058
400E_8180	Port 0 priority resolution configuration (ESW_P0RES)	32	R/W	0000_0000h	43.3.54/ 2059
400E_8184	Port 1 priority resolution configuration (ESW_P1RES)	32	R/W	0000_0000h	43.3.55/ 2060
400E_8188	Port 2 priority resolution configuration (ESW_P2RES)	32	R/W	0000_0000h	43.3.56/ 2061
400E_8200	Port 0 VLAN ID (ESW_P0ID)	32	R/W	0000_0000h	43.3.57/ 2062
400E_8204	Port 1 VLAN ID (ESW_P1ID)	32	R/W	0000_0000h	43.3.58/ 2062
400E_8208	Port 2 VLAN ID (ESW_P2ID)	32	R/W	0000_0000h	43.3.59/ 2063

Table continues on the next page...

ESW memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400E_8280	VLAN domain resolution entry 0 (ESW_VRES0)	32	R/W	0000_0000h	43.3.60/ 2063
400E_8284	VLAN domain resolution entry 1 (ESW_VRES1)	32	R/W	0000_0000h	43.3.61/ 2064
400E_8288	VLAN domain resolution entry 2 (ESW_VRES2)	32	R/W	0000_0000h	43.3.62/ 2065
400E_828C	VLAN domain resolution entry 4 (ESW_VRES3)	32	R/W	0000_0000h	43.3.63/ 2066
400E_8290	VLAN domain resolution entry 4 (ESW_VRES4)	32	R/W	0000_0000h	43.3.64/ 2067
400E_8294	VLAN domain resolution entry 5 (ESW_VRES5)	32	R/W	0000_0000h	43.3.65/ 2068
400E_8298	VLAN domain resolution entry 6 (ESW_VRES6)	32	R/W	0000_0000h	43.3.66/ 2069
400E_829C	VLAN domain resolution entry 7 (ESW_VRES7)	32	R/W	0000_0000h	43.3.67/ 2070
400E_82A0	VLAN domain resolution entry 8 (ESW_VRES8)	32	R/W	0000_0000h	43.3.68/ 2071
400E_82A4	VLAN domain resolution entry 9 (ESW_VRES9)	32	R/W	0000_0000h	43.3.69/ 2072
400E_82A8	VLAN domain resolution entry 10 (ESW_VRES10)	32	R/W	0000_0000h	43.3.70/ 2073
400E_82AC	VLAN domain resolution entry 11 (ESW_VRES11)	32	R/W	0000_0000h	43.3.71/ 2074
400E_82B0	VLAN domain resolution entry 12 (ESW_VRES12)	32	R/W	0000_0000h	43.3.72/ 2075
400E_82B4	VLAN domain resolution entry 13 (ESW_VRES13)	32	R/W	0000_0000h	43.3.73/ 2076
400E_82B8	VLAN domain resolution entry 14 (ESW_VRES14)	32	R/W	0000_0000h	43.3.74/ 2077
400E_82BC	VLAN domain resolution entry 15 (ESW_VRES15)	32	R/W	0000_0000h	43.3.75/ 2078
400E_82C0	VLAN domain resolution entry 16 (ESW_VRES16)	32	R/W	0000_0000h	43.3.76/ 2079
400E_82C4	VLAN domain resolution entry 17 (ESW_VRES17)	32	R/W	0000_0000h	43.3.77/ 2080
400E_82C8	VLAN domain resolution entry 18 (ESW_VRES18)	32	R/W	0000_0000h	43.3.78/ 2081
400E_82CC	VLAN domain resolution entry 19 (ESW_VRES19)	32	R/W	0000_0000h	43.3.79/ 2082
400E_82D0	VLAN domain resolution entry 20 (ESW_VRES20)	32	R/W	0000_0000h	43.3.80/ 2083
400E_82D4	VLAN domain resolution entry 21 (ESW_VRES21)	32	R/W	0000_0000h	43.3.81/ 2084

Table continues on the next page...

ESW memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400E_82D8	VLAN domain resolution entry 22 (ESW_VRES22)	32	R/W	0000_0000h	43.3.82/ 2085
400E_82DC	VLAN domain resolution entry 23 (ESW_VRES23)	32	R/W	0000_0000h	43.3.83/ 2086
400E_82E0	VLAN domain resolution entry 24 (ESW_VRES24)	32	R/W	0000_0000h	43.3.84/ 2087
400E_82E4	VLAN domain resolution entry 25 (ESW_VRES25)	32	R/W	0000_0000h	43.3.85/ 2088
400E_82E8	VLAN domain resolution entry 26 (ESW_VRES26)	32	R/W	0000_0000h	43.3.86/ 2089
400E_82EC	VLAN domain resolution entry 27 (ESW_VRES27)	32	R/W	0000_0000h	43.3.87/ 2090
400E_82F0	VLAN domain resolution entry 28 (ESW_VRES28)	32	R/W	0000_0000h	43.3.88/ 2091
400E_82F4	VLAN domain resolution entry 29 (ESW_VRES29)	32	R/W	0000_0000h	43.3.89/ 2092
400E_82F8	VLAN domain resolution entry 30 (ESW_VRES30)	32	R/W	0000_0000h	43.3.90/ 2093
400E_82FC	VLAN domain resolution entry 31 (ESW_VRES31)	32	R/W	0000_0000h	43.3.91/ 2094
400E_8300	Number of discarded frames (ESW_DISCN)	32	R	0000_0000h	43.3.92/ 2094
400E_8304	Bytes of discarded frames (ESW_DISCB)	32	R	0000_0000h	43.3.93/ 2095
400E_8308	Number of non-discarded frames (ESW_NDISCN)	32	R	0000_0000h	43.3.94/ 2095
400E_830C	Bytes of non-discarded frames (ESW_NDISCB)	32	R	0000_0000h	43.3.95/ 2096
400E_8310	Port 0 output queue congestion (ESW_P0OQC)	32	R	0000_0000h	43.3.96/ 2096
400E_8314	Port 0 mismatching VLAN ID (ESW_P0MVID)	32	R	0000_0000h	43.3.97/ 2097
400E_8318	Port 0 missing VLAN tag (ESW_P0MVTAG)	32	R	0000_0000h	43.3.98/ 2097
400E_831C	Port 0 blocked (ESW_P0BL)	32	R	0000_0000h	43.3.99/ 2098
400E_8320	Port 1 output queue congestion (ESW_P1OQC)	32	R	0000_0000h	43.3.100/ 2098
400E_8324	Port 1 mismatching VLAN ID (ESW_P1MVID)	32	R	0000_0000h	43.3.101/ 2099
400E_8328	Port 1 missing VLAN tag (ESW_P1MVTAG)	32	R	0000_0000h	43.3.102/ 2099
400E_832C	Port 1 blocked (ESW_P1BL)	32	R	0000_0000h	43.3.103/ 2100

Table continues on the next page...

ESW memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400E_8330	Port 2 output queue congestion (ESW_P2OQC)	32	R	0000_0000h	43.3.104/ 2100
400E_8334	Port 2 mismatching VLAN ID (ESW_P2MVID)	32	R	0000_0000h	43.3.105/ 2101
400E_8338	Port 2 missing VLAN tag (ESW_P2MVTAG)	32	R	0000_0000h	43.3.106/ 2101
400E_833C	Port 2 blocked (ESW_P2BL)	32	R	0000_0000h	43.3.107/ 2102
400E_8400	Interrupt status register (ESW_ISR)	32	R/W	0000_0000h	43.3.108/ 2102
400E_8404	Interrupt mask register (ESW_IMR)	32	R/W	0000_0000h	43.3.109/ 2103
400E_8408	Receive descriptor ring pointer (ESW_RDSR)	32	R/W	0000_0000h	43.3.110/ 2105
400E_840C	Transmit descriptor ring pointer (ESW_TDSR)	32	R/W	0000_0000h	43.3.111/ 2106
400E_8410	Maximum receive buffer size (ESW_MRBR)	32	R/W	0000_0000h	43.3.112/ 2106
400E_8414	Receive descriptor active (ESW_RDAR)	32	R/W	0000_0000h	43.3.113/ 2107
400E_8418	Transmit descriptor active (ESW_TDAR)	32	R/W	0000_0000h	43.3.114/ 2108
400E_8500	Learning records A0 & B1 (ESW_LREC0)	32	R	0000_0000h	43.3.115/ 2109
400E_8504	Learning record B1 (ESW_LREC1)	32	R	0000_0000h	43.3.116/ 2110
400E_8508	Learning data available status (ESW_LSR)	32	R	0000_0000h	43.3.117/ 2110

43.3.1 Revision (ESW_REV)

Address: 400E_8000h base + 0h offset = 400E_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSTREV																CORREV															
W																																
Reset	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

* Notes:

- See ESW chip-configuration information for reset value.

ESW_REV field descriptions

Field	Description
31–16 CSTREV	Customer revision number
15–0 CORREV	Core revision number

43.3.2 Scratch register (ESW_SCR)

The scratch register provides a memory location to test register access. It returns all data written to it in inverted form.

Address: 400E_8000h base + 4h offset = 400E_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCRATCH																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_SCR field descriptions

Field	Description
31–0 SCRATCH	Scratch register

43.3.3 Port enable register (ESW_PER)

The port enable register independently enables the transmit and receive direction for each port.

Address: 400E_8000h base + 8h offset = 400E_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved													RE2	RE1	RE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													TE2	TE1	TE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PER field descriptions

Field	Description
31–19 Reserved	This field is reserved. Reserved, must be cleared.
18 RE2	Enable receive on port 2. 0 Disable. The input is ignored and never selected for frame reception. 1 Enable. The port is selected and a frame is accepted if it indicates data available.
17 RE1	Enable receive on port 1. 0 Disable. The input is ignored and never selected for frame reception. 1 Enable. The port is selected and a frame is accepted if it indicates data available.
16 RE0	Enable receive on port 0. 0 Disable. The input is ignored and never selected for frame reception. 1 Enable. The port is selected and a frame is accepted if it indicates data available.
15–3 Reserved	This field is reserved. Reserved, must be cleared.
2 TE2	Enable transmit on port 2. 0 Disable. All frames forwarded to the port are discarded. 1 Enable. A frame can be forwarded to the port.
1 TE1	Enable transmit on port 1. 0 Disable. All frames forwarded to the port are discarded. 1 Enable. A frame can be forwarded to the port.
0 TE0	Enable transmit on port 0. 0 Disable. All frames forwarded to the port are discarded. 1 Enable. A frame can be forwarded to the port.

43.3.4 VLAN verify (ESW_VLANV)

Address: 400E_8000h base + 10h offset = 400E_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved													DU2	DU1	DU0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													VV2	VV1	VV0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VLANV field descriptions

Field	Description
31–19 Reserved	This field is reserved. Reserved, must be cleared.

Table continues on the next page...

ESW_VLANV field descriptions (continued)

Field	Description
18 DU2	Discard unknown. 0 Received frames with unknown VLAN IDs are not discarded 1 Received frames with an unknown VLAN ID or no VLAN tag are discarded and not forwarded (i.e. the default bcast is ignored)
17 DU1	Discard unknown. 0 Received frames with unknown VLAN IDs are not discarded 1 Received frames with an unknown VLAN ID or no VLAN tag are discarded and not forwarded (i.e. the default bcast is ignored)
16 DU0	Discard unknown. 0 Received frames with unknown VLAN IDs are not discarded 1 Received frames with an unknown VLAN ID or no VLAN tag are discarded and not forwarded (i.e. the default bcast is ignored)
15–3 Reserved	This field is reserved. Reserved, must be cleared.
2 VV2	Verify VLAN domain. 0 Frames are routed to the output port without VLAN domain checking 1 A frame is accepted from the port as valid only when the input and output ports are members of the VLAN domain of the frame.
1 VV1	Verify VLAN domain. 0 Frames are routed to the output port without VLAN domain checking 1 A frame is accepted from the port as valid only when the input and output ports are members of the VLAN domain of the frame.
0 VV0	Verify VLAN domain. 0 Frames are routed to the output port without VLAN domain checking 1 A frame is accepted from the port as valid only when the input and output ports are members of the VLAN domain of the frame.

43.3.5 Default broadcast resolution (ESW_DBCR)

The default output port list for broadcast/flooding resolution (see Section 32.5.9.2, "Broadcast/Multicast/VLAN Domain Resolution").

Address: 400E_8000h base + 14h offset = 400E_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_DBCR field descriptions

Field	Description
31–3 Reserved	This field is reserved. Reserved, must be cleared.
2 P2	Default broadcast resolution 0 A frame with the corresponding VLAN ID is not switched to that port 1 Indicates that each port is a member of the VLAN and frames with the corresponding VLAN ID can be switched to the port
1 P1	Default broadcast resolution 0 A frame with the corresponding VLAN ID is not switched to that port 1 Indicates that each port is a member of the VLAN and frames with the corresponding VLAN ID can be switched to the port
0 P0	Default broadcast resolution 0 A frame with the corresponding VLAN ID is not switched to that port 1 Indicates that each port is a member of the VLAN and frames with the corresponding VLAN ID can be switched to the port

43.3.6 Default multicast resolution (ESW_DMCR)

ESW_DMCR is used for broadcast/flooding resolution (see Section 32.5.9.2, "Broadcast/ Multicast/VLAN Domain Resolution").

Address: 400E_8000h base + 18h offset = 400E_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_DMCR field descriptions

Field	Description
31–3 Reserved	This field is reserved. Reserved, must be cleared.
2 P2	Default multicast resolution. When the received frame carries a multicast address, the default output port list used instead of ESW_DBCR.
1 P1	Default multicast resolution. When the received frame carries a multicast address, the default output port list used instead of ESW_DBCR.
0 P0	Default multicast resolution. When the received frame carries a multicast address, the default output port list used instead of ESW_DBCR.

43.3.7 Blocking and learning enable (ESW_BKLR)

ESW_BKLR independently defines the blocking and learning states for each port.

Address: 400E_8000h base + 1Ch offset = 400E_801Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved													LD2	LD1	LD0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													BE2	BE1	BE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_BKLR field descriptions

Field	Description
31–19 Reserved	This field is reserved. Reserved
18 LD2	Disable learning. 0 Enable. 1 Disable. Only bridge protocol data unit frames are learned. Other frames are ignored for learning.
17 LD1	Disable learning. 0 Enable. 1 Disable. Only bridge protocol data unit frames are learned. Other frames are ignored for learning.
16 LD0	Disable learning. 0 Enable. 1 Disable. Only bridge protocol data unit frames are learned. Other frames are ignored for learning.
15–3 Reserved	This field is reserved. Reserved, must be cleared.
2 BE2	Enable blocking. 0 Disable. 1 Enable. Only bridge protocol data units are accepted on that input, all other frames are discarded.
1 BE1	Enable blocking. 0 Disable. 1 Enable. Only bridge protocol data units are accepted on that input, all other frames are discarded.
0 BE0	Enable blocking. 0 Disable. 1 Enable. Only bridge protocol data units are accepted on that input, all other frames are discarded.

43.3.8 Bridge management port configuration (ESW_BMPC)

ESW_BMPC enables and defines the management port that receives bridge protocol frames.

Address: 400E_8000h base + 20h offset = 400E_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved													PORTMASK		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	PRIORITY			Reserved					DIS	EN	MSGTX	Reserved	PORT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_BMPC field descriptions

Field	Description
31–19 Reserved	This field is reserved. Reserved, must be cleared.
18–16 PORTMASK	Portmask for transmission of BPDU frames as defined in Section 32.5.9.5.4, "Management Frame Forwarding". When the management port transmits a BPDU frame to the switch, it is forwarded to all ports in this portmask (bit16=port0, bit17=port1, bit 18=port2).
15–13 PRIORITY	Priority to use for transmitted management frames. Used to e.g. put a management frame in a high-priority output queue for fast delivery.
12–8 Reserved	This field is reserved. Reserved, must be cleared.
7 DIS	If set, BPDU frames are discarded always. Setting has no effect, when the enable bit is set.
6 EN	If set, all BPDU frames are forwarded exclusively to the management port specified in bits 3:0. If cleared, BPDU frames are forwarded as any other frame, or discarded if the discard bit is set.
5 MSGTX	Set (latched) when a BPDU frame as defined in Section 32.5.9.5.4, "Management Frame Forwarding" was transmitted from the management port to any output port. Bit is reset by writing into the register.
4 Reserved	This field is reserved. Reserved, must be cleared.
3–0 PORT	The Port number of the port that should act as a management port. Relevant to all functions that forward frames to the management port (i.e. BPDU processing, snooping). Note: It must be set 0 in this switch configuration (Port 0 to DMA0 is the management port).

43.3.9 Mode configuration (ESW_MODE)

ESW_MODE defines several global configuration settings.

Address: 400E_8000h base + 24h offset = 400E_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	STATRST	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved						P0CT	CRCTRAN	STOP	Reserved					SWEN	SWRST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_MODE field descriptions

Field	Description
31 STATRST	Reset Statistics Counters Command. When set during a write, all statistics counters are cleared. When set, all other bits are ignored and do not influence the currently stored value in the register (i.e. it is not necessary to read/preserve the register contents prior to writing this bit).
30–10 Reserved	This field is reserved. Reserved, must be cleared.
9 P0CT	Enable Port0 input buffer cut-through mode. When cleared (0, default) the input buffer operates in store&forward mode, which is the recommended mode of operation.
8 CRCTRAN	When enabled (1) the MAC ports are expected to process frames to/from the switch including CRC.
7 STOP	Controls toplevel output pin stop_en. No internal function.
6–2 Reserved	This field is reserved. Reserved, must be cleared.
1 SWEN	Controls toplevel output pin switch_en. When deasserted (0), all DMA registers are cleared.
0 SWRST	Controls toplevel output pin switch_reset. No internal function.

43.3.10 VLAN input manipulation select (ESW_VIMSEL)

ESW_VIMSEL defines behavior of the VLAN input manipulation function, if such a function is present on an input port. ESW_VIMSEL has effect only if enabled by the corresponding port bit in ESW_VIMEN.

Address: 400E_8000h base + 28h offset = 400E_8028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																										IM2		IM1		IM0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VIMSEL field descriptions

Field	Description
31–6 Reserved	This field is reserved. Reserved, must be cleared.
5–4 IM2	Input manipulation select for port 2. 00 Mode 1, single tag passthrough 01 Mode 2, single tag overwrite 10 Mode 3, double tag passthrough 11 Mode 4, double tag overwrite
3–2 IM1	Input manipulation select for port 1. 00 Mode 1, single tag passthrough 01 Mode 2, single tag overwrite 10 Mode 3, double tag passthrough 11 Mode 4, double tag overwrite
1–0 IM0	Input manipulation select for port 0. 00 Mode 1, single tag passthrough 01 Mode 2, single tag overwrite 10 Mode 3, double tag passthrough 11 Mode 4, double tag overwrite

43.3.11 VLAN output manipulation select (ESW_VOMSEL)

ESW_VOMSEL defines behavior of the VLAN output manipulation function, if such a function is present on an output port.

Address: 400E_8000h base + 2Ch offset = 400E_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																										OM2		OM1		OM0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VOMSEL field descriptions

Field	Description
31–6 Reserved	This field is reserved. Reserved, must be cleared.
5–4 OM2	Output manipulation select for port 2. 00 No output manipulation 01 Mode 1, strip mode 10 Mode 2, tag through 11 Mode 3, transparent
3–2 OM1	Output manipulation select for port 1. 00 No output manipulation 01 Mode 1, strip mode 10 Mode 2, tag through 11 Mode 3, transparent
1–0 OM0	Output manipulation select for port 0. 00 No output manipulation 01 Mode 1, strip mode 10 Mode 2, tag through 11 Mode 3, transparent

43.3.12 VLAN input manipulation enable (ESW_VIMEN)

ESW_VIMEN enables the input processing according to the ESW_VIMSEL for a port.

Address: 400E_8000h base + 30h offset = 400E_8030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved													EN2	EN1	EN0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VIMEN field descriptions

Field	Description
31–3 Reserved	This field is reserved. Reserved, must be cleared.
2 EN2	Input manipulation enable for port 2. 0 Disable. ESW_VIMSEL has no effect and the frames are processed unmodified. 1 Enable
1 EN1	Input manipulation enable for port 1. 0 Disable. ESW_VIMSEL has no effect and the frames are processed unmodified. 1 Enable
0 EN0	Input manipulation enable for port 0. 0 Disable. ESW_VIMSEL has no effect and the frames are processed unmodified. 1 Enable

43.3.13 VLAN tag ID (ESW_VID)

The VLAN type field value to expect to identify a VLAN-tagged frame. A valid 802.1Q VLAN-tagged frame must use the value 0x8100.

Address: 400E_8000h base + 34h offset = 400E_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	TAG																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

ESW_VID field descriptions

Field	Description
31–0 TAG	ID to identify a VLAN-tagged frame. A valid 802.1Q VLAN-tagged frame must use the value 0x8100.

43.3.14 Mirror control register (ESW_MCR)

The mirror control register defines port mirroring and filtering conditions.

Address: 400E_8000h base + 40h offset = 400E_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved					EGDA	EGSA	INGDA	INGSA	EGMAP	INGMAP	MEN	PORT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_MCR field descriptions

Field	Description
31–11 Reserved	This field is reserved. Reserved, must be cleared.
10 EGDA	If set, only frames transmitted on an egress port with a destination address matching the value in ESW_EGDA{L,H} are mirrored. Other frames are not mirrored. **NOTE:** If the egress map is not enabled (EGMAP = 0) then any frame with a matching destination address is mirrored.
9 EGSA	If set, only frames transmitted on an egress port with a source address matching the value in ESW_EGSA{L,H} are mirrored. Other frames are not mirrored. **NOTE:** If the egress map is not enabled (EGMAP = 0) then any frame with a matching source address is mirrored.
8 INGDA	If set, only frames received on an ingress port with a destination address matching the value in ESW_INGDA{L,H} are mirrored. Other frames are not mirrored. **NOTE:** If the ingress map is not enabled (INGMAP = 0) then any frame with a matching destination address is mirrored.
7 INGSA	If set, only frames received on an ingress port with a source address matching the value in ESW_INGSA{L,H} are mirrored. Other frames are not mirrored.

Table continues on the next page...

ESW_MCR field descriptions (continued)

Field	Description
	NOTE: If the ingress map is not enabled (INGMAP = 0) then any frame with a matching source address is mirrored.
6 EGMAP	Egress map enable. 0 Egress port map has no effect 1 Egress map is enabled. A frame forwarded to an output port that has a bit set in the egress map is mirrored.
5 INGMAP	Ingress map enable. 0 Ingress port map has no effect 1 Ingress map is enabled. A frame received on an ingress port that has a bit set in the ingress map is mirrored.
4 MEN	Mirroring enable. 0 Disabled 1 Enabled
3–0 PORT	The port number that should act as the mirror port and receive all mirrored frames.

43.3.15 Egress port definitions (ESW_EGMAP)

Address: 400E_8000h base + 44h offset = 400E_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													EG2	EG1	EG0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_EGMAP field descriptions

Field	Description
31–3 Reserved	This field is reserved. Reserved, must be cleared.
2 EG2	Port mirroring egress for port 2. 0 Disable 1 Enabled. Frames destined for this port are mirrored to the mirror port.
1 EG1	Port mirroring egress for port 1. 0 Disable 1 Enabled. Frames destined for this port are mirrored to the mirror port.
0 EG0	Port mirroring egress for port 0.

Table continues on the next page...

ESW_EGMAP field descriptions (continued)

Field	Description
	0 Disable 1 Enabled. Frames destined for this port are mirrored to the mirror port.

43.3.16 Ingress port definitions (ESW_INGMAP)

Address: 400E_8000h base + 48h offset = 400E_8048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													ING2	ING1	ING0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_INGMAP field descriptions

Field	Description
31–3 Reserved	This field is reserved. Reserved, must be cleared.
2 ING2	Port mirroring ingress for port 2. 0 Disable 1 Enabled. Frames from this port are mirrored to the mirror port.
1 ING1	Port mirroring ingress for port 1. 0 Disable 1 Enabled. Frames from this port are mirrored to the mirror port.
0 ING0	Port mirroring ingress for port 0. 0 Disable 1 Enabled. Frames from this port are mirrored to the mirror port.

43.3.17 Ingress source MAC address low (ESW_INGSAL)

Address: 400E_8000h base + 4Ch offset = 400E_804Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDLOW																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_INGSAL field descriptions

Field	Description
31–0 ADDLOW	First four bytes of the ingress/egress MAC address for mirror filtering.

43.3.18 Ingress source MAC address high (ESW_INGSAH)

Address: 400E_8000h base + 50h offset = 400E_8050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ADDHIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_INGSAH field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–0 ADDHIGH	First two bytes of the ingress/egress MAC address for mirror filtering.

43.3.19 Ingress destination MAC address low (ESW_INGDAL)

Address: 400E_8000h base + 54h offset = 400E_8054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDLOW																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_INGDAL field descriptions

Field	Description
31–0 ADDLOW	First four bytes of the ingress/egress MAC address for mirror filtering.

43.3.20 Ingress destination MAC address high (ESW_INGDAH)

Address: 400E_8000h base + 58h offset = 400E_8058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ADDHIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_INGDAH field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–0 ADDHIGH	First two bytes of the ingress/egress MAC address for mirror filtering.

43.3.21 Egress source MAC address low (ESW_EGSAL)

Address: 400E_8000h base + 5Ch offset = 400E_805Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDLOW																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_EGSAL field descriptions

Field	Description
31–0 ADDLOW	First four bytes of the ingress/egress MAC address for mirror filtering.

43.3.22 Egress source MAC address high (ESW_EGSAH)

Address: 400E_8000h base + 60h offset = 400E_8060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ADDHIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_EGSAH field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–0 ADDHIGH	First two bytes of the ingress/egress MAC address for mirror filtering.

43.3.23 Egress destination MAC address low (ESW_EGDAL)

Address: 400E_8000h base + 64h offset = 400E_8064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDLOW																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_EGDAL field descriptions

Field	Description
31–0 ADDLOW	First four bytes of the ingress/egress MAC address for mirror filtering.

43.3.24 Egress destination MAC address high (ESW_EGDAH)

Address: 400E_8000h base + 68h offset = 400E_8068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ADDHIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_EGDAH field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–0 ADDHIGH	First two bytes of the ingress/egress MAC address for mirror filtering.

43.3.25 Mirror count value (ESW_MCVAL)

NOTE

If the egress filtering port map is active, every forwarded frame is considered. Otherwise, frames are counted only if the mirroring decision indicated that the frame should be mirrored.

Address: 400E_8000h base + 6Ch offset = 400E_806Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								COUNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_MCVAL field descriptions

Field	Description
31–8 Reserved	This field is reserved. Reserved, must be cleared.
7–0 COUNT	Count value for mirror filtering. Every n^{th} frame is forwarded to the mirror port, if enabled. 0x00 Every frame forwarded 0x01 Every frame forwarded

Table continues on the next page...

ESW_MCVAL field descriptions (continued)

Field	Description
	0x02 Every second frame forwarded ... 0xFF Every 255 th frame forwarded

43.3.26 Memory manager status (ESW_MMSR)

All latched bits are cleared upon a write with any content to the register.

Address: 400E_8000h base + 80h offset = 400E_8080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								CELLS_AVAIL							
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved									DQGRNT	Reserved		MFLATCH	MEMFULL	NOCELL	BUSY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

ESW_MMSR field descriptions

Field	Description
31–24 Reserved	This field is reserved. Reserved, must be cleared.
23–16 CELLS_AVAIL	Real-time indication of currently available cells in memory.
15–7 Reserved	This field is reserved. Reserved, must be cleared.
6 DQGRNT	Indication of if currently inputs are de-queued. Should be set always and is cleared when the memory becomes full (see the MEMFULL bit). **NOTE:** The bit is cleared upon reset. However, set shortly after when the memory manager completes initialization.
5–4 Reserved	This field is reserved. Reserved, must be cleared.
3 MFLATCH	Latched version of mem_full. Is kept set even when mem_full is cleared again. The bit is cleared when the register is written.

Table continues on the next page...

ESW_MMSR field descriptions (continued)

Field	Description
2 MEMFULL	Current memory full indication. The memory is full when less than the programmed minimum cell threshold is available in memory. This is not an error and the memory controller is working fine. It just indicates that the switch does no longer serve its input ports to avoid memory overrun (no_cell error).
1 NOCELL	Set, when memory has exceeded the maximum available number of cells. The event is latched and the bit stays set if the event is no longer active. This is a fatal error and must never happen during operation. The minimum cells threshold must be increased if it happens.The bit is always set after reset (during initialization) and must be cleared when the busy initialization (see bit 0) indication is cleared. **NOTE:** When this bit is set any time during operation (after initialization completed) the switch is in an inoperable state and must be reset completely to restore correct operation. If such an event happens the ESW_LMT setting must be increased during initialization to avoid such situation. The bit is cleared when the register is written.
0 BUSY	When set (1), Memory controller is initializing. The initialization is only preparing the internal data structures within the controller, it does not initialize the shared memory used for frame storage as this is not required. It is asserted after reset and stays set until the memory controller is ready to store frames. The switch must not be enabled before initialization of the memory controller has been completed.

43.3.27 Low memory threshold (ESW_LMT)

If the number of cells available in memory is less than ESW_LMT, the switch discards frames. Choose a value for at least two full-sized frames. A memory overflow due to a too low threshold is a fatal error and may require a device reset.

Address: 400E_8000h base + 84h offset = 400E_8084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								THRESH							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

ESW_LMT field descriptions

Field	Description
31–8 Reserved	This field is reserved. Reserved, must be cleared.
7–0 THRESH	Low memory threshold. The value of this field is the number of 256 bytes. 0x01 0.25 KB (1 x 256 bytes) 0x02 0.5 KB (2 x 256 bytes) ... 0x09 2.25 KB (9 x 256 bytes) ... **NOTE:** Choose a value for at least two full-sized frames.

43.3.28 Lowest number of free cells (ESW_LFC)

ESW_LFC indicates the lowest number of free cells reached in memory during operation since it was last cleared.

Address: 400E_8000h base + 88h offset = 400E_8088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	COUNT																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_LFC field descriptions

Field	Description
31–0 COUNT	Lowest number of free cells reached in memory during operation since it was last cleared. This register is reset to the maximum by writing any value to it.

43.3.29 Port congestion status (ESW_PCSR)

Address: 400E_8000h base + 8Ch offset = 400E_808Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													PC2	PC1	PC0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PCSR field descriptions

Field	Description
31–3 Reserved	This field is reserved. Reserved, must be cleared.
2 PC2	Port congestion status for port 2. 0 Not congested 1 Congested
1 PC1	Port congestion status for port 1. 0 Not congested 1 Congested

Table continues on the next page...

ESW_PCSR field descriptions (continued)

Field	Description
0 PC0	Port congestion status for port 0. 0 Not congested 1 Congested

43.3.30 Switch input and output interface status (ESW_IOSR)

Address: 400E_8000h base + 90h offset = 400E_8090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved													IR2	IR1	IR0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													OR2	OR1	OR0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IOSR field descriptions

Field	Description
31–19 Reserved	This field is reserved. Reserved, must be cleared.
18 IR2	Input data available for port 2. 0 Not available 1 Data available
17 IR1	Input data available for port 1. 0 Not available 1 Data available
16 IR0	Input data available for port 0. 0 Not available 1 Data available
15–3 Reserved	This field is reserved. Reserved, must be cleared.
2 OR2	Output ready to accept data for port 2. 0 Not ready 1 Ready
1 OR1	Output ready to accept data for port 1.

Table continues on the next page...

ESW_IOSR field descriptions (continued)

Field	Description
	0 Not ready 1 Ready
0 OR0	Output ready to accept data for port 0. 0 Not ready 1 Ready

43.3.31 Queue weights (ESW_QWT)

ESW_QWT defines the weight for the corresponding queue for all ports. Setting all weights to zero implements a strict priority scheme.

Address: 400E_8000h base + 94h offset = 400E_8094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			Q3WT					Reserved			Q2WT					Reserved			Q1WT					Reserved			Q0WT				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_QWT field descriptions

Field	Description
31–29 Reserved	This field is reserved. Reserved, must be cleared
28–24 Q3WT	Queue 3 weight. Defines the weight for the corresponding queue for all ports. A higher weight gives more priority to the queue. Queue 3 is the highest priority queue. Valid values are 0-30.
23–21 Reserved	This field is reserved. Reserved, must be cleared
20–16 Q2WT	Queue 2 weight. Defines the weight for the corresponding queue for all ports. A higher weight gives more priority to the queue. Valid values are 0-30.
15–13 Reserved	This field is reserved. Reserved, must be cleared
12–8 Q1WT	Queue 1 weight. Defines the weight for the corresponding queue for all ports. A higher weight gives more priority to the queue. Valid values are 0-30.
7–5 Reserved	This field is reserved. Reserved, must be cleared
4–0 Q0WT	Queue 2 weight. Defines the weight for the corresponding queue for all ports. A higher weight gives more priority to the queue. Valid values are 0-30.

43.3.32 Port 0 Backpressure Congestion Threshold (ESW_P0BCT)

ESW_P0BCT defines the congestion threshold for port 0 backpressure. If the total output queues' shared memory (see Output Frame Queuing) has less than this number of free cells available, the switch stops serving the port 0 input buffer. This eventually fills the input buffer .

Address: 400E_8000h base + 9Ch offset = 400E_809Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								THRESH							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

ESW_P0BCT field descriptions

Field	Description
31–8 Reserved	This field is reserved. Reserved, must be cleared.
7–0 THRESH	Defines the congestion threshold for port 0 backpressure. Clearing this field disables the function (no backpressure). **NOTE:** THRESH must be greater than the memory full threshold, ESW_LMT, to stop the DMA0 before a memory full situation. The recommended value for THRESH is 0x40, which will eliminate or minimize the number of frames discarded in the ESW due to congestion.

43.3.33 Port 0 forced forwarding enable (ESW_FFEN)

ESW_FFEN forces forwarding for port 0 frames (i.e. frames transmitted from the DMA0 to the port 0 of the switch).

Address: 400E_8000h base + BCh offset = 400E_80BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved												FD		Reserved	FEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_FFEN field descriptions

Field	Description
31–4 Reserved	This field is reserved. Reserved, must be cleared.
3–2 FD	When FEN is set, this field defines if the port 0 frame should be forwarded to the MAC at ports 1 and 2. **NOTE:** It is possible to forward to one or both MAC ports. If neither bit is set, FEN is ignored and the frame is processed normally. This can be used to implement a handshake, as FEN is still reset but no further action occurs. 00 Do not forward. Frame is processed normally. 01 Forward to port 1 only 10 Forward to port 2 only 11 Forward to both ports
1 Reserved	This field is reserved. Reserved, must be cleared.
0 FEN	When set, the next frame received from port 0 (the local DMA port) is forwarded to the ports defined in FD. The bit resets to zero automatically when one frame from port 0 has been processed by the switch (i.e. has been read from the port 0 input buffer; see Figure 32-1). Therefore, the bit must be set again as necessary. See alsoSection 32.5.8.2, "Forced Forwarding" for a description.

43.3.34 Port snooping registers (ESW_PSNP1)

ESW_PSNP1 defines the TCP/UDP port number snooping function configuration.

Address: 400E_8000h base + C0h offset = 400E_80C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PORT_COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											CS	CD	MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PSNP1 field descriptions

Field	Description
31–16 PORT_ COMPARE	The 16-bit port number to compare within the TCP or UDP header of a frame. Note: it is possible to set both the compare source/dest bits. The result is OR'ed, meaning if any of the fields match the compare value, the frame is processed as defined by the mode bits.
15–5 Reserved	This field is reserved. Reserved, must be cleared.
4 CS	When set, the TCP or UDP source port number field within the frame is compared with the compare value provided in 31:16.
3 CD	When set, the TCP or UDP destination port number field within the frame is compared with the compare value provided in PORT_COMPARE.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value. **NOTE:** The management port is defined in ESW_BMPC register. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the TCP/UDP destination port value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the snooping function only if the enable bit is set. Otherwise the settings are ignored. When written with 0 will also force bits 3,4 to 0. Defaults to 0 upon reset.

43.3.35 Port snooping registers (ESW_PSNP2)

ESW_PSNP2 defines the TCP/UDP port number snooping function configuration.

Address: 400E_8000h base + C4h offset = 400E_80C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	PORT_COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved											CS	CD	MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PSNP2 field descriptions

Field	Description
31–16 PORT_ COMPARE	The 16-bit port number to compare within the TCP or UDP header of a frame. Note: it is possible to set both the compare source/dest bits. The result is OR'ed, meaning if any of the fields match the compare value, the frame is processed as defined by the mode bits.
15–5 Reserved	This field is reserved. Reserved, must be cleared.
4 CS	When set, the TCP or UDP source port number field within the frame is compared with the compare value provided in 31:16.
3 CD	When set, the TCP or UDP destination port number field within the frame is compared with the compare value provided in PORT_COMPARE.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value. **NOTE:** The management port is defined in ESW_BMPC register. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the TCP/UDP destination port value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the snooping function only if the enable bit is set. Otherwise the settings are ignored. When written with 0 will also force bits 3,4 to 0. Defaults to 0 upon reset.

43.3.36 Port snooping registers (ESW_PSNP3)

ESW_PSNP3 defines the TCP/UDP port number snooping function configuration.

Address: 400E_8000h base + C8h offset = 400E_80C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PORT_COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											CS	CD	MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PSNP3 field descriptions

Field	Description
31–16 PORT_ COMPARE	The 16-bit port number to compare within the TCP or UDP header of a frame. Note: it is possible to set both the compare source/dest bits. The result is OR'ed, meaning if any of the fields match the compare value, the frame is processed as defined by the mode bits.
15–5 Reserved	This field is reserved. Reserved, must be cleared.
4 CS	When set, the TCP or UDP source port number field within the frame is compared with the compare value provided in 31:16.
3 CD	When set, the TCP or UDP destination port number field within the frame is compared with the compare value provided in PORT_COMPARE.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value. **NOTE:** The management port is defined in ESW_BMPC register. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the TCP/UDP destination port value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the snooping function only if the enable bit is set. Otherwise the settings are ignored. When written with 0 will also force bits 3,4 to 0. Defaults to 0 upon reset.

43.3.37 Port snooping registers (ESW_PSNP4)

ESW_PSNP4 defines the TCP/UDP port number snooping function configuration.

Address: 400E_8000h base + CCh offset = 400E_80CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	PORT_COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved											CS	CD	MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PSNP4 field descriptions

Field	Description
31–16 PORT_ COMPARE	The 16-bit port number to compare within the TCP or UDP header of a frame. Note: it is possible to set both the compare source/dest bits. The result is OR'ed, meaning if any of the fields match the compare value, the frame is processed as defined by the mode bits.
15–5 Reserved	This field is reserved. Reserved, must be cleared.
4 CS	When set, the TCP or UDP source port number field within the frame is compared with the compare value provided in 31:16.
3 CD	When set, the TCP or UDP destination port number field within the frame is compared with the compare value provided in PORT_COMPARE.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value. **NOTE:** The management port is defined in ESW_BMPC register. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the TCP/UDP destination port value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the snooping function only if the enable bit is set. Otherwise the settings are ignored. When written with 0 will also force bits 3,4 to 0. Defaults to 0 upon reset.

43.3.38 Port snooping registers (ESW_PSNP5)

ESW_PSNP5 defines the TCP/UDP port number snooping function configuration.

Address: 400E_8000h base + D0h offset = 400E_80D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PORT_COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											CS	CD	MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PSNP5 field descriptions

Field	Description
31–16 PORT_ COMPARE	The 16-bit port number to compare within the TCP or UDP header of a frame. Note: it is possible to set both the compare source/dest bits. The result is OR'ed, meaning if any of the fields match the compare value, the frame is processed as defined by the mode bits.
15–5 Reserved	This field is reserved. Reserved, must be cleared.
4 CS	When set, the TCP or UDP source port number field within the frame is compared with the compare value provided in 31:16.
3 CD	When set, the TCP or UDP destination port number field within the frame is compared with the compare value provided in PORT_COMPARE.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value. **NOTE:** The management port is defined in ESW_BMPC register. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the TCP/UDP destination port value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the snooping function only if the enable bit is set. Otherwise the settings are ignored. When written with 0 will also force bits 3,4 to 0. Defaults to 0 upon reset.

43.3.39 Port snooping registers (ESW_PSNP6)

ESW_PSNP6 defines the TCP/UDP port number snooping function configuration.

Address: 400E_8000h base + D4h offset = 400E_80D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PORT_COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											CS	CD	MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PSNP6 field descriptions

Field	Description
31–16 PORT_ COMPARE	The 16-bit port number to compare within the TCP or UDP header of a frame. Note: it is possible to set both the compare source/dest bits. The result is OR'ed, meaning if any of the fields match the compare value, the frame is processed as defined by the mode bits.
15–5 Reserved	This field is reserved. Reserved, must be cleared.
4 CS	When set, the TCP or UDP source port number field within the frame is compared with the compare value provided in 31:16.
3 CD	When set, the TCP or UDP destination port number field within the frame is compared with the compare value provided in PORT_COMPARE.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value. **NOTE:** The management port is defined in ESW_BMPC register. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the TCP/UDP destination port value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the snooping function only if the enable bit is set. Otherwise the settings are ignored. When written with 0 will also force bits 3,4 to 0. Defaults to 0 upon reset.

43.3.40 Port snooping registers (ESW_PSNP7)

ESW_PSNP7 defines the TCP/UDP port number snooping function configuration.

Address: 400E_8000h base + D8h offset = 400E_80D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PORT_COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											CS	CD	MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PSNP7 field descriptions

Field	Description
31–16 PORT_ COMPARE	The 16-bit port number to compare within the TCP or UDP header of a frame. Note: it is possible to set both the compare source/dest bits. The result is OR'ed, meaning if any of the fields match the compare value, the frame is processed as defined by the mode bits.
15–5 Reserved	This field is reserved. Reserved, must be cleared.
4 CS	When set, the TCP or UDP source port number field within the frame is compared with the compare value provided in 31:16.
3 CD	When set, the TCP or UDP destination port number field within the frame is compared with the compare value provided in PORT_COMPARE.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value. **NOTE:** The management port is defined in ESW_BMPC register. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the TCP/UDP destination port value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the snooping function only if the enable bit is set. Otherwise the settings are ignored. When written with 0 will also force bits 3,4 to 0. Defaults to 0 upon reset.

43.3.41 Port snooping registers (ESW_PSNP8)

ESW_PSNP8 defines the TCP/UDP port number snooping function configuration.

Address: 400E_8000h base + DCh offset = 400E_80DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PORT_COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											CS	CD	MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_PSNP8 field descriptions

Field	Description
31–16 PORT_ COMPARE	The 16-bit port number to compare within the TCP or UDP header of a frame. Note: it is possible to set both the compare source/dest bits. The result is OR'ed, meaning if any of the fields match the compare value, the frame is processed as defined by the mode bits.
15–5 Reserved	This field is reserved. Reserved, must be cleared.
4 CS	When set, the TCP or UDP source port number field within the frame is compared with the compare value provided in 31:16.
3 CD	When set, the TCP or UDP destination port number field within the frame is compared with the compare value provided in PORT_COMPARE.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value. **NOTE:** The management port is defined in ESW_BMPC register. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the TCP/UDP destination port value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the snooping function only if the enable bit is set. Otherwise the settings are ignored. When written with 0 will also force bits 3,4 to 0. Defaults to 0 upon reset.

43.3.42 IP snooping registers (ESW_IPSNP1)

ESW_IPSNP1 defines the IP snooping function configuration.

Address: 400E_8000h base + E0h offset = 400E_80E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PROTOCOL								Reserved					MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IPSNP1 field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–8 PROTOCOL	The 8-bit protocol value to match with the incoming frame's IP header protocol field.
7–3 Reserved	This field is reserved. Reserved, must be cleared.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value (see the PROTOTCOL bits). **NOTE:** The management port is defined in register ESW_BMPC. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the protocol value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the IP snooping function only if the enable bit is set. Otherwise the settings are ignored. Defaults to 0 upon reset.

43.3.43 IP snooping registers (ESW_IPSNP2)

ESW_IPSNP2 defines the IP snooping function configuration.

Address: 400E_8000h base + E4h offset = 400E_80E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PROTOCOL								Reserved					MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IPSNP2 field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–8 PROTOCOL	The 8-bit protocol value to match with the incoming frame's IP header protocol field.
7–3 Reserved	This field is reserved. Reserved, must be cleared.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value (see the PROTOTCOL bits). **NOTE:** The management port is defined in register ESW_BMPC. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the protocol value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the IP snooping function only if the enable bit is set. Otherwise the settings are ignored. Defaults to 0 upon reset.

43.3.44 IP snooping registers (ESW_IPSNP3)

ESW_IPSNP3 defines the IP snooping function configuration.

Address: 400E_8000h base + E8h offset = 400E_80E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PROTOCOL								Reserved					MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IPSNP3 field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–8 PROTOCOL	The 8-bit protocol value to match with the incoming frame's IP header protocol field.
7–3 Reserved	This field is reserved. Reserved, must be cleared.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value (see the PROTOTCOL bits). **NOTE:** The management port is defined in register ESW_BMPC. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the protocol value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the IP snooping function only if the enable bit is set. Otherwise the settings are ignored. Defaults to 0 upon reset.

43.3.45 IP snooping registers (ESW_IPSNP4)

ESW_IPSNP4 defines the IP snooping function configuration.

Address: 400E_8000h base + ECh offset = 400E_80ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PROTOCOL								Reserved					MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IPSNP4 field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–8 PROTOCOL	The 8-bit protocol value to match with the incoming frame's IP header protocol field.
7–3 Reserved	This field is reserved. Reserved, must be cleared.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value (see the PROTOTCOL bits). **NOTE:** The management port is defined in register ESW_BMPC. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the protocol value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the IP snooping function only if the enable bit is set. Otherwise the settings are ignored. Defaults to 0 upon reset.

43.3.46 IP snooping registers (ESW_IPSNP5)

ESW_IPSNP5 defines the IP snooping function configuration.

Address: 400E_8000h base + F0h offset = 400E_80F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	PROTOCOL								Reserved					MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IPSNP5 field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–8 PROTOCOL	The 8-bit protocol value to match with the incoming frame's IP header protocol field.
7–3 Reserved	This field is reserved. Reserved, must be cleared.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value (see the PROTOTCOL bits). **NOTE:** The management port is defined in register ESW_BMPC. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the protocol value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the IP snooping function only if the enable bit is set. Otherwise the settings are ignored. Defaults to 0 upon reset.

43.3.47 IP snooping registers (ESW_IPSNP6)

ESW_IPSNP6 defines the IP snooping function configuration.

Address: 400E_8000h base + F4h offset = 400E_80F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PROTOCOL								Reserved					MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IPSNP6 field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–8 PROTOCOL	The 8-bit protocol value to match with the incoming frame's IP header protocol field.
7–3 Reserved	This field is reserved. Reserved, must be cleared.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value (see the PROTOTCOL bits). **NOTE:** The management port is defined in register ESW_BMPC. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the protocol value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the IP snooping function only if the enable bit is set. Otherwise the settings are ignored. Defaults to 0 upon reset.

43.3.48 IP snooping registers (ESW_IPSNP7)

ESW_IPSNP7 defines the IP snooping function configuration.

Address: 400E_8000h base + F8h offset = 400E_80F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PROTOCOL								Reserved					MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IPSNP7 field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–8 PROTOCOL	The 8-bit protocol value to match with the incoming frame's IP header protocol field.
7–3 Reserved	This field is reserved. Reserved, must be cleared.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value (see the PROTOTCOL bits). **NOTE:** The management port is defined in register ESW_BMPC. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the protocol value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the IP snooping function only if the enable bit is set. Otherwise the settings are ignored. Defaults to 0 upon reset.

43.3.49 IP snooping registers (ESW_IPSNP8)

ESW_IPSNP8 defines the IP snooping function configuration.

Address: 400E_8000h base + FCh offset = 400E_80FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PROTOCOL								Reserved					MODE		EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IPSNP8 field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–8 PROTOCOL	The 8-bit protocol value to match with the incoming frame's IP header protocol field.
7–3 Reserved	This field is reserved. Reserved, must be cleared.
2–1 MODE	Defines the forwarding that should occur, when an IP frame is received and the protocol field matches the protocol value (see the PROTOTCOL bits). **NOTE:** The management port is defined in register ESW_BMPC. 00 Forward frame to designated management port only 01 Copy to management port and forward normally 10 Discard 11 Reserved
0 EN	When set (1) the entry contains valid data and the function is active. If a match with the protocol value occurs, the frame is processed as defined by the mode setting. All other bits of the register are interpreted by the IP snooping function only if the enable bit is set. Otherwise the settings are ignored. Defaults to 0 upon reset.

43.3.50 Port 0 VLAN priority resolution map (ESW_P0VRES)

The ESW_P0VRES register implements a 3-bit to 3-bit VLAN priority mapping capability. The current frame's 3-bit VLAN priority field is used as an index and the corresponding priority is taken from the respective position of the register giving the final classification for the frame.

Address: 400E_8000h base + 100h offset = 400E_8100h

Bit	31–24	23–21	20–18	17–15	14–12	11–9	8–6	5–3	2–0
R/W	Reserved	PRI7	PRI6	PRI5	PRI4	PRI3	PRI2	PRI1	PRI0
Reset	0 0 0 0 0 0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0

ESW_P0VRES field descriptions

Field	Description
31–24 Reserved	This field is reserved. Reserved, must be cleared.
23–21 PRI7	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI7 field to give the final priority classification for the frame.
20–18 PRI6	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI6 field to give the final priority classification for the frame.
17–15 PRI5	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI5 field to give the final priority classification for the frame.
14–12 PRI4	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI4 field to give the final priority classification for the frame.
11–9 PRI3	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI3 field to give the final priority classification for the frame.
8–6 PRI2	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI2 field to give the final priority classification for the frame.
5–3 PRI1	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI1 field to give the final priority classification for the frame.
2–0 PRI0	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI0 field to give the final priority classification for the frame.

43.3.51 Port 1 VLAN priority resolution map (ESW_P1VRES)

The ESW_P1VRES register implements a 3-bit to 3-bit VLAN priority mapping capability. The current frame's 3-bit VLAN priority field is used as an index and the corresponding priority is taken from the respective position of the register giving the final classification for the frame.

Address: 400E_8000h base + 104h offset = 400E_8104h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PRI7			PRI6			PRI5			PRI4			PRI3			PRI2			PRI1			PRI0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P1VRES field descriptions

Field	Description
31–24 Reserved	This field is reserved. Reserved, must be cleared.
23–21 PRI7	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI7 field to give the final priority classification for the frame.
20–18 PRI6	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI6 field to give the final priority classification for the frame.
17–15 PRI5	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI5 field to give the final priority classification for the frame.
14–12 PRI4	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI4 field to give the final priority classification for the frame.
11–9 PRI3	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI3 field to give the final priority classification for the frame.
8–6 PRI2	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI2 field to give the final priority classification for the frame.
5–3 PRI1	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI1 field to give the final priority classification for the frame.
2–0 PRI0	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI0 field to give the final priority classification for the frame.

43.3.52 Port 2 VLAN priority resolution map (ESW_P2VRES)

The ESW_P2VRES register implements a 3-bit to 3-bit VLAN priority mapping capability. The current frame's 3-bit VLAN priority field is used as an index and the corresponding priority is taken from the respective position of the register giving the final classification for the frame.

Address: 400E_8000h base + 108h offset = 400E_8108h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PRI7			PRI6			PRI5			PRI4			PRI3			PRI2			PRI1			PRI0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P2VRES field descriptions

Field	Description
31–24 Reserved	This field is reserved. Reserved, must be cleared.
23–21 PRI7	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI7 field to give the final priority classification for the frame.
20–18 PRI6	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI6 field to give the final priority classification for the frame.
17–15 PRI5	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI5 field to give the final priority classification for the frame.
14–12 PRI4	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI4 field to give the final priority classification for the frame.
11–9 PRI3	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI3 field to give the final priority classification for the frame.
8–6 PRI2	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI2 field to give the final priority classification for the frame.
5–3 PRI1	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI1 field to give the final priority classification for the frame.
2–0 PRI0	The current frame's 3-bit VLAN priority field is an index to the corresponding PRI0 field to give the final priority classification for the frame.

43.3.53 IPv4/v6 priority resolution table (ESW_IPRES)

Address: 400E_8000h base + 140h offset = 400E_8140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	READ	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	PRI2		PRI1		PRI0		IPV4SEL	ADDRESS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IPRES field descriptions

Field	Description
31 READ	Must be cleared to write values in the tables. When set during register writes, the IPv6 select and address bits are stored in the register only and the priority bits are ignored and not written into the addressed table. When the register is read, the priority bits represent the value read from the table always.
30–15 Reserved	This field is reserved. Reserved, must be cleared.
14–13 PRI2	The priority information to write into the addressed table entry. These 2 bits represent the output priority selected when the frame is received on port 2. When reading from the register, the bits show the value from the addressed table entry (address from last write operation).
12–11 PRI1	The priority information to write into the addressed table entry. These 2 bits represent the output priority selected when the frame is received on port 1. When reading from the register, the bits show the value from the addressed table entry (address from last write operation).
10–9 PRI0	The priority information to write into the addressed table entry. These 2 bits represent the output priority selected when the frame is received on port 0. 00=priority 0 (will be forwarded to output queue 0) 01=priority 1 (output queue 1) 10=priority 2 (output queue 2) 11=priority 3 (output queue 3) When reading from the register, the bits show the value from the addressed table entry (address from last write operation).
8 IPV4SEL	If set during a write, the IPv4 table is accessed. Valid address values range from 0 to 63. If cleared, the IPv6 table is accessed. Valid address values range from 0 to 255.
7–0 ADDRESS	The address of the priority entry to read or write for a frame received on port n. The IPv4 priority table has 64 entries. The IPv6 table has 256 entries. See also Section 32.5.5.2.1, "Classification Table Programming Model" for a description of the mapping table.

43.3.54 Port 0 priority resolution configuration (ESW_P0RES)

ESW_P0RES defines which priority information should be used for priority resolution.

Address: 400E_8000h base + 180h offset = 400E_8180h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved									DFLT_PRI			Reserved	MAC	IP	VLAN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P0RES field descriptions

Field	Description
31–7 Reserved	This field is reserved. Reserved, must be cleared.
6–4 DFLT_PRI	The default priority of a frame received on port n, if none of the priority resolutions could define a priority of the frame. Up to 3 bits can be implemented.
3 Reserved	This field is reserved. Reserved, must be cleared.
2 MAC	Enable MAC based priority resolution for frame received on port n. If set, the priority information found within the MAC address table is used.
1 IP	Enable IP priority resolution for frame received on port n. If set, the IP DiffServ/COS field is used and priority is resolved according to the ESW_IPRES setting for the port. If cleared, IP Diffserv/COS fields are ignored.
0 VLAN	Enable VLAN priority resolution for frame received on port n. If set, the VLAN tag field of a frame is inspected and priority is resolved according to the setting programmed in ESW_P *n* VRES for the port on which the frame was received. If cleared, VLAN priority is ignored.

43.3.55 Port 1 priority resolution configuration (ESW_P1RES)

ESW_P1RES defines which priority information should be used for priority resolution.

Address: 400E_8000h base + 184h offset = 400E_8184h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved									DFLT_PRI			Reserved	MAC	IP	VLAN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P1RES field descriptions

Field	Description
31–7 Reserved	This field is reserved. Reserved, must be cleared.
6–4 DFLT_PRI	The default priority of a frame received on port n, if none of the priority resolutions could define a priority of the frame. Up to 3 bits can be implemented.
3 Reserved	This field is reserved. Reserved, must be cleared.
2 MAC	Enable MAC based priority resolution for frame received on port n. If set, the priority information found within the MAC address table is used.
1 IP	Enable IP priority resolution for frame received on port n. If set, the IP DiffServ/COS field is used and priority is resolved according to the ESW_IPRES setting for the port. If cleared, IP Diffserv/COS fields are ignored.
0 VLAN	Enable VLAN priority resolution for frame received on port n. If set, the VLAN tag field of a frame is inspected and priority is resolved according to the setting programmed in ESW_P *n* VRES for the port on which the frame was received. If cleared, VLAN priority is ignored.

43.3.56 Port 2 priority resolution configuration (ESW_P2RES)

ESW_P2RES defines which priority information should be used for priority resolution.

Address: 400E_8000h base + 188h offset = 400E_8188h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved									DFLT_PRI			Reserved	MAC	IP	VLAN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P2RES field descriptions

Field	Description
31–7 Reserved	This field is reserved. Reserved, must be cleared.
6–4 DFLT_PRI	The default priority of a frame received on port n, if none of the priority resolutions could define a priority of the frame. Up to 3 bits can be implemented.
3 Reserved	This field is reserved. Reserved, must be cleared.
2 MAC	Enable MAC based priority resolution for frame received on port n. If set, the priority information found within the MAC address table is used.
1 IP	Enable IP priority resolution for frame received on port n. If set, the IP DiffServ/COS field is used and priority is resolved according to the ESW_IPRES setting for the port. If cleared, IP Diffserv/COS fields are ignored.
0 VLAN	Enable VLAN priority resolution for frame received on port n. If set, the VLAN tag field of a frame is inspected and priority is resolved according to the setting programmed in ESW_P *n* VRES for the port on which the frame was received. If cleared, VLAN priority is ignored.

43.3.57 Port 0 VLAN ID (ESW_P0ID)

ESW_P0ID defines the VLAN ID field for VLAN input manipulation function of a port (if it exists).

Address: 400E_8000h base + 200h offset = 400E_8200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																VLANID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P0ID field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–0 VLANID	VLAN ID field for the VLAN input manipulation function.

43.3.58 Port 1 VLAN ID (ESW_P1ID)

ESW_P1ID defines the VLAN ID field for VLAN input manipulation function of a port (if it exists).

Address: 400E_8000h base + 204h offset = 400E_8204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																VLANID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P1ID field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–0 VLANID	VLAN ID field for the VLAN input manipulation function.

43.3.59 Port 2 VLAN ID (ESW_P2ID)

ESW_P2ID defines the VLAN ID field for VLAN input manipulation function of a port (if it exists).

Address: 400E_8000h base + 208h offset = 400E_8208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																VLANID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P2ID field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved, must be cleared.
15–0 VLANID	VLAN ID field for the VLAN input manipulation function.

43.3.60 VLAN domain resolution entry 0 (ESW_VRES0)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 280h offset = 400E_8280h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES0 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.

Table continues on the next page...

ESW_VRES0 field descriptions (continued)

Field	Description
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.61 VLAN domain resolution entry 1 (ESW_VRES1)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 284h offset = 400E_8284h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES1 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.62 VLAN domain resolution entry 2 (ESW_VRES2)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 288h offset = 400E_8288h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES2 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.63 VLAN domain resolution entry 4 (ESW_VRES3)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 28Ch offset = 400E_828Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES3 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.64 VLAN domain resolution entry 4 (ESW_VRES4)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 290h offset = 400E_8290h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES4 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.65 VLAN domain resolution entry 5 (ESW_VRES5)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 294h offset = 400E_8294h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES5 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.66 VLAN domain resolution entry 6 (ESW_VRES6)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 298h offset = 400E_8298h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES6 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.67 VLAN domain resolution entry 7 (ESW_VRES7)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 29Ch offset = 400E_829Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES7 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.68 VLAN domain resolution entry 8 (ESW_VRES8)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2A0h offset = 400E_82A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES8 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.69 VLAN domain resolution entry 9 (ESW_VRES9)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2A4h offset = 400E_82A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES9 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.70 VLAN domain resolution entry 10 (ESW_VRES10)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2A8h offset = 400E_82A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES10 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.71 VLAN domain resolution entry 11 (ESW_VRES11)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2ACh offset = 400E_82ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES11 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.72 VLAN domain resolution entry 12 (ESW_VRES12)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2B0h offset = 400E_82B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES12 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.73 VLAN domain resolution entry 13 (ESW_VRES13)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2B4h offset = 400E_82B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES13 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.74 VLAN domain resolution entry 14 (ESW_VRES14)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2B8h offset = 400E_82B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES14 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.75 VLAN domain resolution entry 15 (ESW_VRES15)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2BCh offset = 400E_82BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES15 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.76 VLAN domain resolution entry 16 (ESW_VRES16)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2C0h offset = 400E_82C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES16 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.77 VLAN domain resolution entry 17 (ESW_VRES17)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2C4h offset = 400E_82C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES17 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.78 VLAN domain resolution entry 18 (ESW_VRES18)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2C8h offset = 400E_82C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES18 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.79 VLAN domain resolution entry 19 (ESW_VRES19)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2CCh offset = 400E_82CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES19 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.80 VLAN domain resolution entry 20 (ESW_VRES20)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2D0h offset = 400E_82D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES20 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.81 VLAN domain resolution entry 21 (ESW_VRES21)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2D4h offset = 400E_82D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES21 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.82 VLAN domain resolution entry 22 (ESW_VRES22)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2D8h offset = 400E_82D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES22 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.83 VLAN domain resolution entry 23 (ESW_VRES23)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2DCh offset = 400E_82DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES23 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.84 VLAN domain resolution entry 24 (ESW_VRES24)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2E0h offset = 400E_82E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES24 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.85 VLAN domain resolution entry 25 (ESW_VRES25)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2E4h offset = 400E_82E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES25 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.86 VLAN domain resolution entry 26 (ESW_VRES26)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2E8h offset = 400E_82E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES26 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.87 VLAN domain resolution entry 27 (ESW_VRES27)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2ECh offset = 400E_82ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES27 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.88 VLAN domain resolution entry 28 (ESW_VRES28)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2F0h offset = 400E_82F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES28 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.89 VLAN domain resolution entry 29 (ESW_VRES29)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2F4h offset = 400E_82F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES29 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.90 VLAN domain resolution entry 30 (ESW_VRES30)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2F8h offset = 400E_82F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES30 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.91 VLAN domain resolution entry 31 (ESW_VRES31)

NOTE

The VLAN table is always searched completely. Therefore, the table entries do not need to be written in any order.

Address: 400E_8000h base + 2FCh offset = 400E_82FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	VLANID												P2	P1	P0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_VRES31 field descriptions

Field	Description
31–15 Reserved	This field is reserved. Reserved, must be cleared.
14–3 VLANID	VLAN identifier
2 P2	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
1 P1	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.
0 P0	One bit per port that is member of the VLAN identified with the 12-bit VLAN ID of the entry.

43.3.92 Number of discarded frames (ESW_DISCN)

Total number of incoming frames processed but discarded in the switch

Address: 400E_8000h base + 300h offset = 400E_8300h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_DISCN field descriptions

Field	Description
31–0 COUNT	Total number of incoming frames processed but discarded in the switch

43.3.93 Bytes of discarded frames (ESW_DISCB)

Sum of bytes of frames counted in ESW_DISCN

Address: 400E_8000h base + 304h offset = 400E_8304h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_DISCB field descriptions

Field	Description
31–0 COUNT	Sum of bytes of frames counted in ESW_DISCN

43.3.94 Number of non-discarded frames (ESW_NDISCN)

Total number of incoming frames processed and not discarded

Address: 400E_8000h base + 308h offset = 400E_8308h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_NDISCN field descriptions

Field	Description
31–0 COUNT	Total number of incoming frames processed and not discarded

43.3.95 Bytes of non-discarded frames (ESW_NDISCB)

Sum of bytes of frames counted in ESW_NDISCN

Address: 400E_8000h base + 30Ch offset = 400E_830Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_NDISCB field descriptions

Field	Description
31–0 COUNT	Sum of bytes of frames counted in ESW_NDISCN

43.3.96 Port 0 output queue congestion (ESW_P0OQC)

Port 0 outgoing frames discarded due to output queue congestion

Address: 400E_8000h base + 310h offset = 400E_8310h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P0OQC field descriptions

Field	Description
31–0 COUNT	Number of outgoing frames discarded due to output queue congestion

43.3.97 Port 0 mismatching VLAN ID (ESW_P0MVID)

Port 0 incoming frames discarded due to mismatching or missing VLAN ID while VLAN verification was enabled. See ESW_VLANV.

Address: 400E_8000h base + 314h offset = 400E_8314h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P0MVID field descriptions

Field	Description
31–0 COUNT	Number of incoming frames discarded due to mismatching or missing VLAN ID while VLAN verification was enabled

43.3.98 Port 0 missing VLAN tag (ESW_P0MVTAG)

Port 0 incoming frames discarded due to missing VLAN tag. See ESW_VLANV.

Address: 400E_8000h base + 318h offset = 400E_8318h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P0MVTAG field descriptions

Field	Description
31–0 COUNT	Number of incoming frames discarded due to missing VLAN tag

43.3.99 Port 0 blocked (ESW_P0BL)

Port 0 incoming frames discarded (after learning) as port is configured in blocking mode

Address: 400E_8000h base + 31Ch offset = 400E_831Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P0BL field descriptions

Field	Description
31–0 COUNT	Number of incoming frames discarded (after learning) as port is configured in blocking mode

43.3.100 Port 1 output queue congestion (ESW_P1OQC)

Port 1 outgoing frames discarded due to output queue congestion

Address: 400E_8000h base + 320h offset = 400E_8320h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P1OQC field descriptions

Field	Description
31–0 COUNT	Number of outgoing frames discarded due to output queue congestion

43.3.101 Port 1 mismatching VLAN ID (ESW_P1MVID)

Port 1 incoming frames discarded due to mismatching or missing VLAN ID while VLAN verification was enabled. See ESW_VLANV.

Address: 400E_8000h base + 324h offset = 400E_8324h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P1MVID field descriptions

Field	Description
31–0 COUNT	Number of incoming frames discarded due to mismatching or missing VLAN ID while VLAN verification was enabled

43.3.102 Port 1 missing VLAN tag (ESW_P1MVTAG)

Port 1 incoming frames discarded due to missing VLAN tag. See ESW_VLANV.

Address: 400E_8000h base + 328h offset = 400E_8328h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P1MVTAG field descriptions

Field	Description
31–0 COUNT	Number of incoming frames discarded due to missing VLAN tag

43.3.103 Port 1 blocked (ESW_P1BL)

Port 1 incoming frames discarded (after learning) as port is configured in blocking mode

Address: 400E_8000h base + 32Ch offset = 400E_832Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P1BL field descriptions

Field	Description
31–0 COUNT	Number of incoming frames discarded (after learning) as port is configured in blocking mode

43.3.104 Port 2 output queue congestion (ESW_P2OQC)

Port 2 outgoing frames discarded due to output queue congestion

Address: 400E_8000h base + 330h offset = 400E_8330h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P2OQC field descriptions

Field	Description
31–0 COUNT	Number of outgoing frames discarded due to output queue congestion

43.3.105 Port 2 mismatching VLAN ID (ESW_P2MVID)

Port 2 incoming frames discarded due to mismatching or missing VLAN ID while VLAN verification was enabled. See ESW_VLANV.

Address: 400E_8000h base + 334h offset = 400E_8334h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P2MVID field descriptions

Field	Description
31–0 COUNT	Number of incoming frames discarded due to mismatching or missing VLAN ID while VLAN verification was enabled

43.3.106 Port 2 missing VLAN tag (ESW_P2MVTAG)

Port 2 incoming frames discarded due to missing VLAN tag. See ESW_VLANV.

Address: 400E_8000h base + 338h offset = 400E_8338h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P2MVTAG field descriptions

Field	Description
31–0 COUNT	Number of incoming frames discarded due to missing VLAN tag

43.3.107 Port 2 blocked (ESW_P2BL)

Port 2 incoming frames discarded (after learning) as port is configured in blocking mode

Address: 400E_8000h base + 33Ch offset = 400E_833Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_P2BL field descriptions

Field	Description
31–0 COUNT	Number of incoming frames discarded (after learning) as port is configured in blocking mode

43.3.108 Interrupt status register (ESW_ISR)

ESW_ISR indicates the interrupt status. To clear a bit write a one to it. The bit stays set if the event condition persists.

Address: 400E_8000h base + 400h offset = 400E_8400h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						LRN	OD2	OD1	OD0	QM	TXF	TXB	RXF	RXB	EBERR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_ISR field descriptions

Field	Description
31–10 Reserved	This field is reserved. Reserved, must be cleared.
9 LRN	Learning Record available in registers LNR_REC_0 and LNR_REC_1 . Note: this interrupt can be very frequent on a heavy loaded network. It is not recommended to use this interrupt source as interrupt but rather implement a slow background task polling the bit to perform learning.
8 OD2	Outgoing frames discarded due to output Queue congestion on Port 2 or port is disabled (ESW_PER).
7 OD1	Outgoing frames discarded due to output Queue congestion on Port 1or port is disabled (ESW_PER).
6 OD0	Outgoing frames discarded due to output Queue congestion on Port 0 or port is disabled (ESW_PER).
5 QM	Low Memory Threshold. Asserted if the memory became congested and number of free cells dropped below threshold ESW_LMT . **NOTE:** This field will become asserted after reset immediately due to memory initialization.
4 TXF	Transmit frame interrupt. This bit indicates a frame has been transmitted and the last corresponding buffer descriptor has been updated .
3 TXB	Transmit buffer interrupt. This bit indicates a transmit buffer descriptor has been updated .
2 RXF	Receive frame interrupt. This bit indicates a frame has been received and the last corresponding buffer descriptor has been updated .
1 RXB	Receive buffer interrupt. This bit indicates a receive buffer descriptor not the last in the frame has been updated .
0 EBERR	Ethernet bus error. This bit indicates a system bus error occurs when a DMA transaction is underway .

43.3.109 Interrupt mask register (ESW_IMR)

Address: 400E_8000h base + 404h offset = 400E_8404h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved						LRN	OD2	OD1	OD0	QM	TXF	TXB	RXF	RXB	EBERR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_IMR field descriptions

Field	Description
31–10 Reserved	This field is reserved. Reserved, must be cleared.
9 LRN	Ths bit corresponds to the ESW_ISR[LRN] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[LRN] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .
8 OD2	Ths bit corresponds to the ESW_ISR[OD2] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[OD2] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .
7 OD1	Ths bit corresponds to the ESW_ISR[OD1] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[OD1] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .
6 OD0	Ths bit corresponds to the ESW_ISR[OD0] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[OD0] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .
5 QM	Ths bit corresponds to the ESW_ISR[QM] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[QM] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .
4 TXF	Ths bit corresponds to the ESW_ISR[TXF] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[TXF] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .
3 TXB	Ths bit corresponds to the ESW_ISR[TXB] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[TXB] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .
2 RXF	Ths bit corresponds to the ESW_ISR[RXF] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[RXF] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .

Table continues on the next page...

ESW_IMR field descriptions (continued)

Field	Description
1 RXB	Ths bit corresponds to the ESW_ISR[RXB] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[RXB] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .
0 EBERR	Ths bit corresponds to the ESW_ISR[EBERR] interrupt source and determines whether the interrupt condition can generate an interrupt. At each processor clock, ESW_ISR samples the signal generated by the interrupting source. ESW_ISR[EBERR] reflects the state of the interrupt signal even if this bit is masked. 0 The interrupt source is masked 1 The interrupt source is not masked and an interrupt can occur .

43.3.110 Receive descriptor ring pointer (ESW_RDSR)

ESW_RDSR points to the start of the circular receive buffer descriptor queue in external memory. This pointer must be 32-bit aligned; however, it is recommended it be made 128-bit aligned (evenly divisible by 16).This register is not reset and must be initialized prior to operation.

Address: 400E_8000h base + 408h offset = 400E_8408h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_RDSR field descriptions

Field	Description
31–2 ADDRESS	Pointer to start of receive buffer descriptor queue.
1–0 Reserved	This field is reserved. Reserved, must be cleared.

43.3.111 Transmit descriptor ring pointer (ESW_TDSR)

ESW_TSDR provides a pointer to the start of the circular transmit buffer descriptor queue in external memory. This pointer must be 32-bit aligned; however, it is recommended it be made 128-bit aligned (evenly divisible by 16). You should write zeros to bits 1 and 0. Hardware ignores non-zero values in these two bit positions.

This register is undefined at reset and must be initialized prior to operation.

Address: 400E_8000h base + 40Ch offset = 400E_840Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_TDSR field descriptions

Field	Description
31–2 ADDRESS	Pointer to start of transmit buffer descriptor queue.
1–0 Reserved	This field is reserved. Reserved, must be cleared.

43.3.112 Maximum receive buffer size (ESW_MRBR)

The ESW_MRBR dictates the maximum size of all receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer. To allow one maximum size frame per buffer, ESW_MRBR must be set to ENET *n* _RCR[MAX_FL] or larger. To properly align the buffer, ESW_MRBR must be evenly divisible by 16. To ensure this, bits 3-0 are forced low.

To minimize bus utilization (descriptor fetches), it is recommended that ESW_MRBR be greater than or equal to 256 bytes.

The ESW_MRBR register is undefined at reset and must be initialized by the user.

Address: 400E_8000h base + 410h offset = 400E_8410h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																		SIZE										Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_MRBR field descriptions

Field	Description
31–14 Reserved	This field is reserved. Reserved, must be cleared.
13–4 SIZE	Maximum size of receive buffer size in bytes. To minimize bus utilization (descriptor fetches), set this field to 256 bytes (0x10) or larger. 0x010 256 + 15 bytes (minimum size recommended) 0x011 272 + 15 bytes ... 0x3FF 16,368 + 15 bytes
3–0 Reserved	This field is reserved. Reserved, must be cleared.

43.3.113 Receive descriptor active (ESW_RDAR)

ESW_RDAR is a command register, written by the user, indicating the receive descriptor ring is updated (the driver produced empty receive buffers with the empty bit set).

When the register is written, the RDAR bit is set. This is independent of the data actually written by the user. When set, the FEC polls the receive descriptor ring and processes receive frames (provided ENET *n* _ECR[ETHER_EN] is also set). After the MAC polls a receive descriptor whose empty bit is not set, the MAC clears the RDAR bit and ceases receive descriptor ring polling until the user sets the bit again, signifying that additional descriptors are placed into the receive descriptor ring.

The ESW_RDAR register is cleared at reset and when ENET *n* _ECR[ETHER_EN] is cleared.

Address: 400E_8000h base + 414h offset = 400E_8414h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved							RDAR	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_RDAR field descriptions

Field	Description
31–25 Reserved	This field is reserved. Reserved, must be cleared.
24 RDAR	Set to 1 when this register is written, regardless of the value written. Cleared by the FEC device when no additional empty descriptors remain in the receive ring. Also cleared when ENET *n* _ECR[ETHER_EN] is cleared.
23–0 Reserved	This field is reserved. Reserved, must be cleared.

43.3.114 Transmit descriptor active (ESW_TDAR)

The ENET *n* _TDAR are command registers which the user writes to indicate the transmit descriptor ring is updated (transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor).

When the register is written, the TDAR bit is set. This value is independent of the data actually written by the user. When set, the MAC polls the transmit descriptor ring and processes transmit frames (provided ENET *n* _ECR[ETHER_EN] is also set). After the MAC polls a transmit descriptor that is a ready bit not set, MAC clears TDAR and ceases transmit descriptor ring polling until you set the bit again, signifying additional descriptors are placed into the transmit descriptor ring.

The ENETn_TDAR registers are cleared at reset, when ENET *n* _ECR[ETHER_EN] is cleared, or when ENETn_ECR[RESET] is set.

Address: 400E_8000h base + 418h offset = 400E_8418h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved							TDAR	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_TDAR field descriptions

Field	Description
31–25 Reserved	This field is reserved. Reserved, must be cleared.
24 TDAR	Set to 1 when this register is written, regardless of the value written. Cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ENET *n* _ECR[ETHER_EN] is cleared.
23–0 Reserved	This field is reserved. Reserved, must be cleared.

43.3.115 Learning records A0 & B1 (ESW_LREC0)

ESW_LREC0 must be read first, followed by reading ESW_LREC1.

Address: 400E_8000h base + 500h offset = 400E_8500h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAC_ADDR0																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_LREC0 field descriptions

Field	Description
31–0 MAC_ADDR0	Lower 32-Bit of the Frame MAC Address. 7:0 = first octet, 31:24=4th octet. Note: this register must be read first, before reading ESW_LREC1

43.3.116 Learning record B1 (ESW_LREC1)

ESW_LREC0 must be read first, followed by reading ESW_LREC1.

Address: 400E_8000h base + 504h offset = 400E_8504h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						SWPORT		HASH							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAC_ADDR1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_LREC1 field descriptions

Field	Description
31–26 Reserved	This field is reserved. Reserved, must be cleared.
25–24 SWPORT	Port number on which the Frame is received.
23–16 HASH	The 8-bit Hash value
15–0 MAC_ADDR1	Upper 16-Bit of the Frame MAC Address. 7:0=5th octet, 15:8=6th octet.

43.3.117 Learning data available status (ESW_LSR)

Address: 400E_8000h base + 508h offset = 400E_8508h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															DA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESW_LSR field descriptions

Field	Description
31–1 Reserved	This field is reserved. Reserved, must be cleared.
0 DA	Indicates if the learning record is valid and can be read. 0 Learning record invalid 1 Learning record valid

43.4 MAC address lookup table

The MAC Address Lookup Table contains 2048 MAC address lookup entries. Each entry is 64 bits. The table is implemented as 32-bit registers, so each 64-bit entry is split into two 32-bit words. The low address (ADDRL) represents the lower 32 bits (31:0). The high address (ADDRH) represents the higher 32 bits (63:32) of an entry. See Address Memory for the structure of a 64-bit entry.

Each entry must be written or read with the low address accessed first followed by the high address. The table should be initialized by software during system startup.

MAC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	MAC Address Lookup Entry Low (MAC_ADDRL)	32	R/W	Undefined	43.4.1/2111
4	MAC Address Lookup Entry High (MAC_ADDRH)	32	R/W	Undefined	43.4.2/2112

43.4.1 MAC Address Lookup Entry Low (MAC_ADDRL)

Address: 0h base + 0h offset = 0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LOW32																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MAC_ADDRL*n* field descriptions

Field	Description
31–0 LOW32	Lower 32 bits of MAC address lookup entry This register must be written or read before the corresponding ADDRH register.

43.4.2 MAC Address Lookup Entry High (MAC_ADDRH)

Address: 0h base + 4h offset = 4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HIGH32																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MAC_ADDRH*n* field descriptions

Field	Description
31–0 HIGH32	Higher 32 bits of MAC address lookup entry This register must be written or read after the corresponding ADDRL register.

43.5 Functional Description

The switch implements the following main functions:

- Input/output VLAN processing
- IP snooping
- Input frame parsing and priority extraction
- Input port selection
- Output port(s) resolution
- Frame queuing
- Output queue scheduling

43.5.1 VLAN Input Processing Function

The VLAN input processing function is used on each switch input port to inspect and manipulate the VLAN tag of frames entering the switch. It performs the following functions:

- Input frame parsing
- VLAN tag insertion or manipulation

Based on the information of the input processing function the frame can be switched to the corresponding output port or is discarded.

43.5.1.1 Terms and Definitions

- VLAN information — The 16-bit field following the VLAN type field within a frame
- VLAN ID — The lower 12 bits of the VLAN information field
- VLAN priority — The upper 3 bits of the VLAN information field that prioritizes incoming frames. A value of 0 represents lowest priority; a value of 7 represents highest priority.

43.5.1.2 Configuration Information

The switch management provides the following information to configure and control the operation of the function:

- ESW_P*n*ID — 16 bit value. The VLAN information field (VLAN-ID and priority) used for tag insertion operations.
- Mode of operation —There are different modes of operation, which define how incoming frames must be processed for a port. The function can be enabled and configured individually per port. See the ESW_VIMEN and ESW_VIMSEL registers.

Note

If the VLAN input processing function is not enabled (ESW_VIMEN = 0) the mode setting has no effect.

43.5.1.3 Modes of Operation

43.5.1.3.1 Frame Processing

The VLAN input processing function modifies the frames before they enter the switching engine. If a VLAN tag is inserted, the switch only acts on the inserted VLAN tag (e.g. priority). Any original tag that was found in the frame before the modification, if any, has no effect within the switch.

In addition, if VLAN verification is enabled for a port (see the ESW_VLANV register), the VLAN ID used for insertion (ESW_P*n*ID) must also be configured in the global VLAN resolution table (see the ESW_VRES*n* register). This ensures the switch accepts frames, which contain the inserted tag.

When a tag is inserted in any of the modes, it is always inserted as the first tag (outer) and its information field is set as programmed in the ESW_P*n*ID register for the port *n* where the frame is received.

43.5.1.3.2 Mode 1 — Single Tagging with Passthrough

Mode 1 inserts a tag only if the frame is untagged. If the frame is already tagged, the frame is unmodified.

43.5.1.3.3 Mode 2 — Single Tagging with Replace

If untagged, add the tag. If single tagged, overwrite it.

43.5.1.3.4 Mode 3 — Double Tagging with Passthrough

Insert a tag on untagged and tagged frames. This results in a single-tagged frame when an untagged is received, and a double-tagged frame, when a single-tagged frame is received. When a double-tagged frame is received, the frame is unmodified.

43.5.1.3.5 Mode 4 — Double Tagging with Replace

Insert tag on untagged and single-tagged frames. If a double-tagged frame is received, overwrite the outer tag.

43.5.2 IP Snooping

The switch supports programmable snooping for up to eight programmable IP protocols. If the protocol field of an IPv4 or IPv6 frame matches one of the programmed values and snooping is enabled for that entry, the frame can be processed as follows:

- Forward to designated management port only
- Copy to management port and normal forward/flood
- Discard on match

The management port is identified by the port number set in the ESW_BMPC register. The function is configured using ESW_IPSNP1–8.

The snooping function can be enabled/disabled individually for each of the entries. If no protocol matches, a match occurs but snooping is disabled, or the frame is coming from the management port itself, the frame is processed normally.

Note

Snooping respects any optional VLAN tags (i.e. extracts next after last VLAN tag).

43.5.3 TCP/UDP Port Number Snooping

Programmable snooping for up to eight programmable TCP or UDP port numbers. If the source or destination port number field within an TCP/IP or UDP/IP frame (IPv4 and IPv6) matches the compare value and snooping is enabled for that entry, the frame can be processed as follows:

- Forward to designated management port only
- Copy to management port and normal forward/flood
- Discard on match

The management port is identified by the port number set in the ESW_BMPC register. The function is configured using ESW_IPSNP1–8.

The snooping function can be enabled/disabled individually for each of the entries. If no entry matches, a match occurs but snooping is disabled, or the frame is coming from the management port itself, the frame is processed normally.

Note

Port number snooping is possible only if the IP header ends up to ten words (40 bytes) after the MAC header. If the IP header ends later (e.g. IPv6 + VLAN or IPv4 + >20 byte options) the port numbers cannot be parsed any more and the port number snooping is ignored (protocol-based snooping is not affected by this limit).

For IPv6 frames the port number can only be compared if the UDP or TCP header is the very next header to the IPv6 header (i.e. it does not detect such headers if any extension headers are present in an IPv6 frame before the TCP or UDP header).

43.5.4 VLAN Output Processing Function

The VLAN output processing function is used on a switch output port to manipulate the VLAN tag of the outgoing frames that leave the switch. Frames are processed based on the output processing mode and the number of tags in the frame.

43.5.4.1 Configuration Information

The switch management provides the information on operating mode to configure and control the operation of the function using the ESW_VOMSEL register of a port. There are three different modes of operation, which define how the outgoing frames should be processed.

43.5.4.1.1 Mode 0 — Disabled

No frame manipulation occurs.

43.5.4.1.2 Mode 1 — Strip Mode

In strip mode, all the tags (single or double) are removed from incoming frame.

43.5.4.1.3 Mode 2 — Tag Through Mode

In tag through mode, the inner tag is passed through while the outer tag is removed for a double-tagged frame. The following rules apply:

- When a single-tagged frame is received, strip the tag from the frame.
- When a double-tagged frame is received, strip the outer tag from the frame.

43.5.4.1.4 Mode 3 — Transparent Mode

In transparent mode, a single-tagged frame is unchanged. The following rules apply:

- When a single-tagged frame is received, frame is unchanged.
- When a double-tagged frame is received, strip the outer tag from the frame.

43.5.5 Frame Classification and Priority Resolution

When a frame is received on an input port, several pieces are extracted from the frame (Ethernet MAC address, VLAN tag, and IP headers) to determine the frame type and perform the relevant classification actions.

In addition, the MAC address table can provide a priority indication for the destination MAC address if the switch management has programmed the address table accordingly (static entry).

The frame is classified in up to four priority levels (0 = lowest, 3 = highest) and is eventually queued in the corresponding priority queue at the output port.

43.5.5.1 VLAN Priority Look-Up

An eight-entry programmable priority table is implemented on each port. The ESW_P*n*VRES registers contain the priority mapping for port *n*.

The switch uses 3-bit priority field from the VLAN tag information to extract the corresponding bits from the table, which indicates which priority the frame is finally classified.

The index in the mapping table is the three bits of the first octet of the VLAN tag data (bit 5 (prio0) is the lsb and bit 7 (prio2) is the msb).

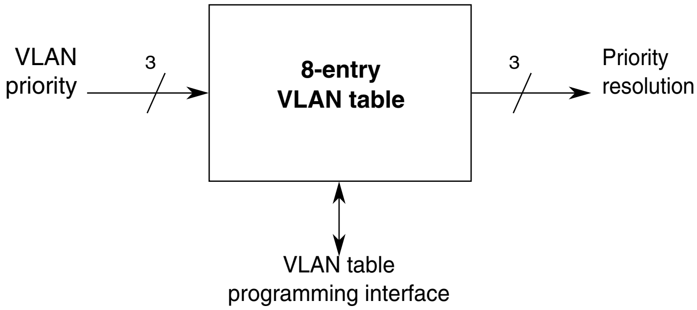


Figure 43-123. VLAN Table Overview

43.5.5.2 IPv4 and IPv6 Priority Look Up

The switch can classify IPv4 and IPv6 frames:

- A 64-entry table is implemented per port to classify the IPv4 frames
  - The frame's six-bit DiffServ field is provided and the table returns the 3-bit priority information
- A 256-entry table is implemented per port to classify IPv6 Frames (IP COS tables)
  - The eight-bit class of service field is provided and the table returns the 3-bit priority information

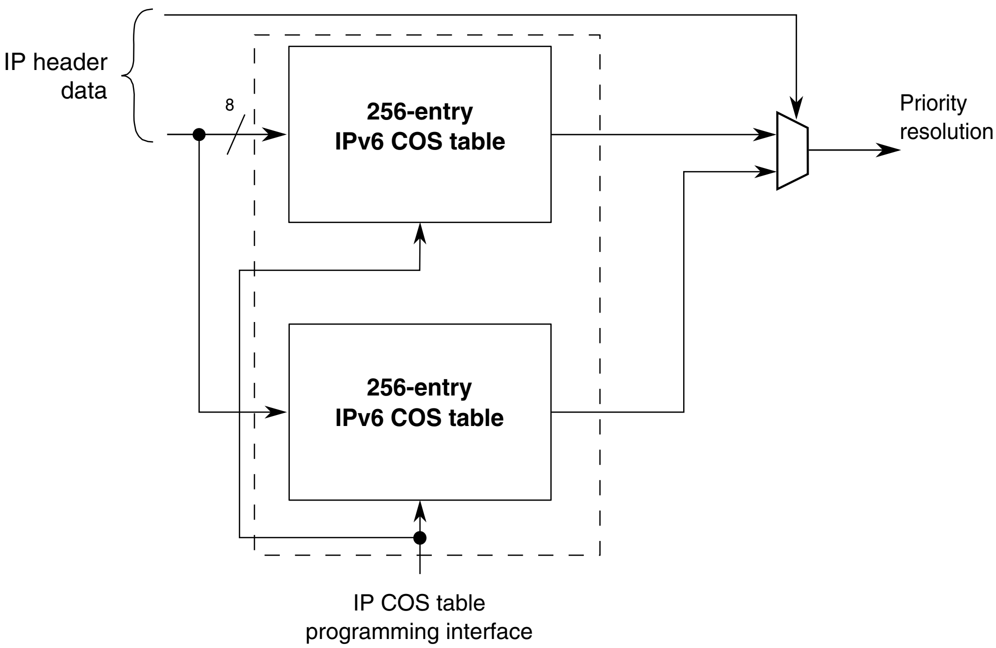


Figure 43-124. IP COS Tables Overview

43.5.5.2.1 Classification Table Programming Model

An indirect addressing scheme is implemented to program the mapping tables using a single register (ESW_IPRES).

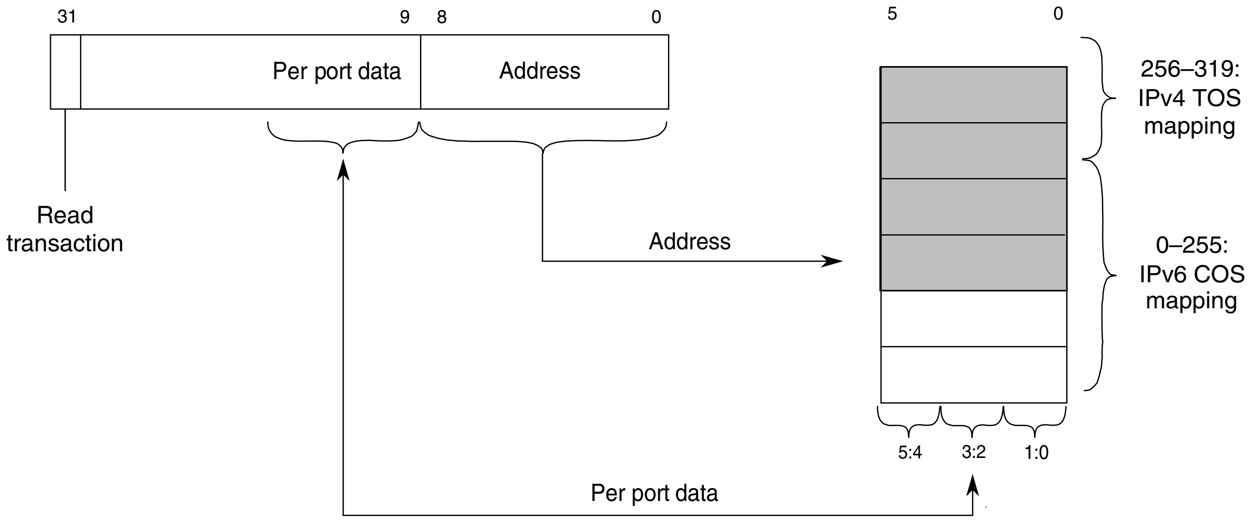


Figure 43-125. ESW_IPRES Mapping Table Programming Model

The table is implemented in a single 320-deep table used for IPv4 6-bit TOS and IPv6 8-bit COS mappings.

- The first 256 entries represent the IPv6 COS field mapping. The received COS field of a frame is used to address a row from 0–255. The value stored is read and used as priority for the frame.
- The last 64 entries represent the IPv4 DiffServ field mapping. The received DiffServ (upper 6-bits of TOS) value of a frame is used to address a row from 256–319.
- Each entry of the table provides the priority mapping in bits 1:0 for port 0 (00 = queue0, 01 = queue1, 10 = queue2, 11 = queue3), bits 3:2 for port 1, and bits 5:4 for port 2.

To write a table row into the table the address is provided in bits 8:0 and the data in bits 13:9, where bit 9 represents bit 0 of the data and bit 13 represents bit 5 of the data.

To read a table row, the read bit must be set when writing into the ESW_IPRES register. This triggers a read transaction to the address provided in bits 8:0 of the register. After this, reading the register provides the data returned from the table for this address.

When writing an entry into the table, software can only write the mapping for all ports in one write transaction. Therefore software must implement a read-modify-write scheme, to:

1. Read the current table contents
2. Modify the priority bits for the port of interest without modifying the other ports bits
3. Write back the complete data word into the table

43.5.5.3 Priority Resolution

The priority resolution function is independently programmable on each port through the ESW_P*n*RES registers by enabling or disabling VLAN, IP, or MAC address-based classification (see Port 0 priority resolution configuration (ESW_P0RES)).

The priority resolution follows the following rule set depending on which classifications are enabled (ESW_P*n*RES) and which fields are found within the frame:

- If IP classification is enabled and an IP header found, map the priority according to the ESW_IPRES table
- Else, if VLAN classification is enabled and a VLAN tag is found, map the priority according to the ESW_P*n*VRES table
- Else, if MAC classification is enabled and MAC address found, take the priority from address table, if it is a static entry
- Else, use default priority as specified in ESW_P*n*RES

43.5.5.4 Bridge Control Protocol Identification

To allow for implementation of bridge control protocols like the spanning tree protocol, all control frames (bridge protocol data units, BPDU) are marked when they enter the switch. The mark then can be used by the input port blocking function to drop the frame after the address learning (see Protocol Snooping).

In addition, the function can be configured to pass all frames or to pass only control frames (e.g. covering spanning tree port states blocking, listening, and learning) and discard all other frames.

43.5.6 Input Port Selection

The port selection constantly polls all input ports for available data. If any data is available, the port is selected and frame data is read from the input. After one frame is read, another port is selected, even if more data is available on the current port.

This means for the application on a port atlantic input interface, that it is not allowed to perform back-to-back frame transfers to the switch. Instead the application must wait for a new selection after one frame has been transferred.

43.5.7 Layer 2 Look-Up Engine

A hash code is calculated using the frame destination's MAC address. It is used as an entry (address) to a table, which contains MAC addresses with destination port number and validity information.

As one hash code value can represent more than one MAC address, the memory implements for each pointer, up to eight MAC address entries, which are searched linearly.

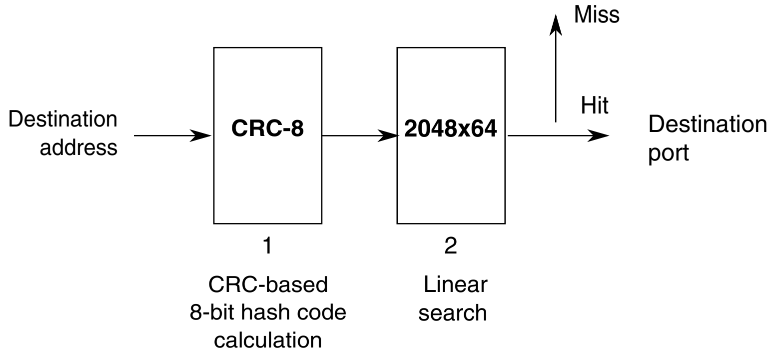


Figure 43-126. Port Look-Up Overview

43.5.7.1 Hash Code

For a MAC address table up to 2048 entries, an 8-bit hash value is calculated from the 48-bit destination MAC address. The hash code uses a CRC-8:

$x^8 + x^2 + x + 1$ (0x07)

43.5.7.2 Address Memory

The address memory is divided into blocks. Each block contains eight records, which contain 64 bits of information each. Each record contains the 48-bit MAC address and provides the necessary forwarding information, and priority or timestamp information.

Two types of records are defined:

- Dynamic record — The dynamic entry provides the MAC address together with a 10-bit timestamp and destination port number. These entries are created by the learning function based on received frames to enable forwarding of frames to dedicated ports. Dynamic entries are deleted by the aging function if not updated.
- Static multiport/priority record — Switch management can also write static entries in the table, which can include priority information and multiple destination ports for forwarding. The MAC address can be unicast or multicast. These records can be used to specify the ports to participate in a specific multicast domain or to assign a MAC address based priority to a frame. The aging and learning functions ignore static records.

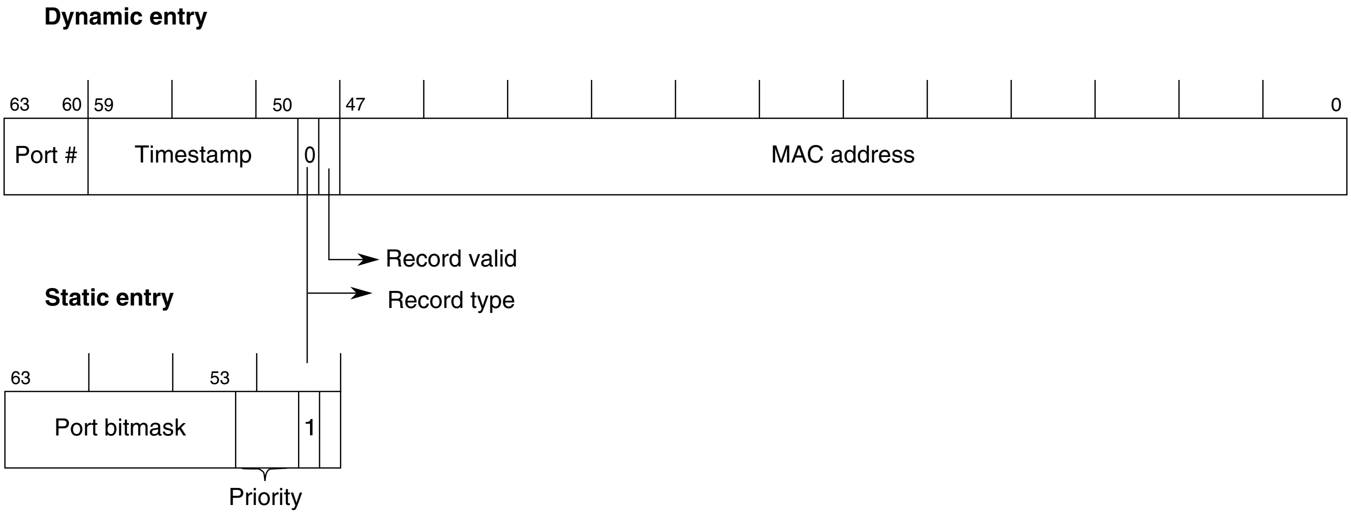


Figure 43-127. Address Memory Record Types

The record's bit 49 decides which type of record is found in the table:

- If 0, a dynamic entry is available. The 10-bit timestamp and 4-bit port number are given in the upper bits of the record.
- If 1, a static entry is available with a 3-bit priority field followed by a 3-bit port bit mask. The record bit 53 represents port 0, bit 54 port 1, and bit 55 port 2. The frame is forwarded to all ports whose port bitmask is set. The source port is removed dynamically from the bitmask during forwarding (a frame is never forwarded to the port where it came from).

The 48-bit MAC address is stored with the first octet in bits 7:0 and the sixth octet in 47:40 of the record.

43.5.8 Layer 2 Lookup Tasks Overview

43.5.8.1 MAC Address Lookup

The 48-bit destination MAC address of each frame received by the switch, on any of its interfaces, is extracted by the hardware and provided to the look-up engine together with the physical interface number. If the frame carries a VLAN tag, the tag information is also extracted and provided to the look-up engine.

If the received frame is a unicast or multicast frame, a two-stage lookup process is implemented. The look-up engine first calculates the hash value from the MAC address. The hash code is used as an entry to the switch address table. The look-up function can provide three results with tree different associated actions performed by the switch hardware:

1. The address is in the table and associated with a correct port number:

   The switch forwards the frame only to the looked up port.

2. The address is in the table but is associated with the port on which it was received:

   The switch discards the frame and does not forward it to any port.

3. The address is not found in the table:

   The switch engine sends the received frame to all ports except the port on which it was received (flooding).

If a broadcast frame is received, the switch hardware sends the received frame to all output ports, except the one from which it was received (flooding).

Note

Flooding and additional frame filtering can be controlled (for example, to avoid the duplication of critical information to unwanted destinations) with the mechanism described in Broadcast/Multicast/VLAN Domain Resolution.

43.5.8.2 Forced Forwarding

The MAC address lookup result can be overwritten using the forced forwarding configuration available in the ESW_FFEN register. This feature is available only for frames coming from the local port (port 0).

When forced forwarding is enabled for a frame, the frame is forwarded to the forced destination ports, ignoring any results from the MAC destination address lookup. Forced forwarding only replaces the MAC lookup function, all other filtering functions (e.g. VLAN verification) act as normal.

43.5.8.3 Learning

The switch hardware extracts the source MAC address of each frame received on each of the switch ports and provides it (via the ESW_LREC0 and ESW_LREC1 registers) to the switch firmware which implements the learning task. The ESW_LSR register indicates availability of learning data.

43.5.8.3.1 Learning Interface

The interface implements a FIFO buffer that stores up to 32 words of 32-bit each.

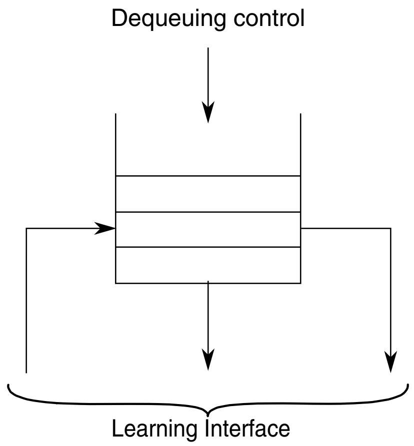


Figure 43-128. Learning Interface Overview

For each frame processed by the switch engine, two 32-bit records (record A0 and record B1) are written in the information FIFO. Record A is written first.

The MAC address available in records A0 and B1 is the source MAC address of the frame. Record A holds the first four bytes of the frame source address, and record B contains the last two bytes.

The hash code in record B is calculated with the source MAC address and the same hash polynomial used for look-up, as defined in Hash Code.

The 4-bit port number defines the port/MAC address association.

Table 43-123. Frame Information Records — 8-Bit Hash Values

	3 3 2 2	2 2 2 2	2 2 2 2	1 1 1 1	1 1 1 1	1 1		
	1 0 9 8	7 6 5 4	3 2 1 0	9 8 7 6	5 4 3 2	1 0 9 8	7 6 5 4	3 2 1 0
Frame record A	MAC address							
Frame record B	Reserved	Port #	Hash code		MAC address (cont'd)			

When information for at least one frame (two records) is available, the status indication in ESW_LSR register is set. To read the frame records, read the ESW_LREC0 register (record A) first followed by the ESW_LREC1 register (record B).

Note

Reading ESW_LREC1 triggers the retrieval of the next record pair from the FIFO, if any.

The learning task (software) uses this information and then executes as follows:

1. For every frame received, the source address with port and timestamp information is stored in the address lookup table. The following information is stored for each entry:
   - MAC address
   - Time stamp: a 10-bit value, determines the age of an entry
   - Port number: a 4-bit value, indicates the port the frame was received
2. If the MAC address table is full, a new entry replaces the oldest entry with an identical matching hash value.

43.5.8.4 Migration

If the firmware receives a MAC address, which is already in the switch table but is associated to a different physical port number, the current entry is overwritten with the new information and the timestamp is set to the current time.

43.5.8.5 Aging

Aging refers to deleting old entries within the table. It proceeds as follows:

1. The 10-bit timestamp is stored with all the entries and is updated each time the source address appears.

2. If a record is not updated for a longer period of time, it is removed from the table if it is a dynamic entry.
3. Static entries are not affected by the aging process and are always kept.

This process runs continuously in the background of the firmware. The aging period is software-controlled and programmable, defaulting to four seconds per step. This gives a range of 4 seconds to 68 minutes.

43.5.9 Frame-Forwarding Tasks

When an input port is selected the frame is forwarded to its corresponding output port. Output port resolution and switching is based on the information from the two-stage MAC address look-up (see Layer 2 Lookup Tasks Overview) followed by additional resolution functions to allow frame duplication and flooding control. These are described in the following sections.

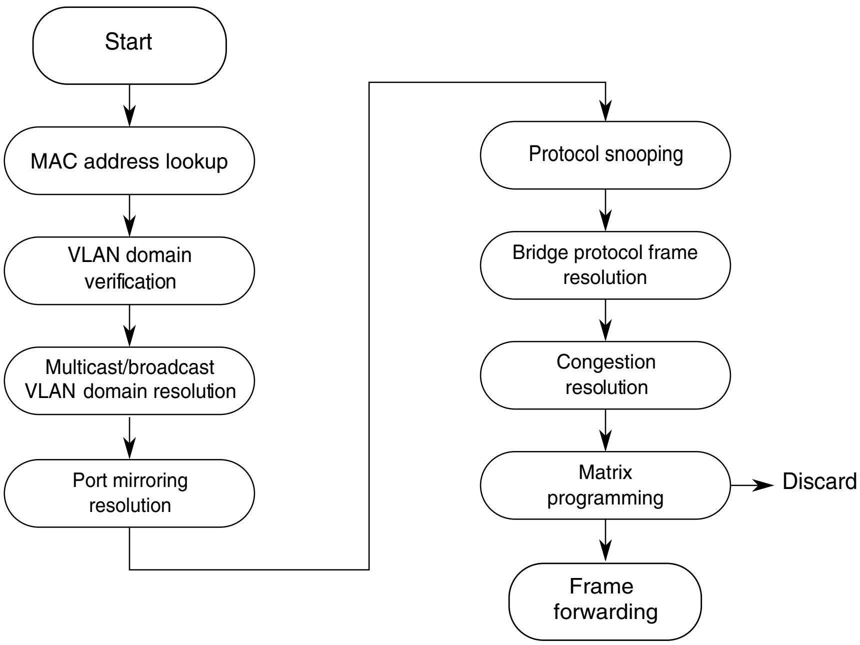


Figure 43-129. Frame Forwarding Tasks Overview

43.5.9.1 VLAN Domain Verification

When the L2 MAC address lookup is successful and identifies a dedicated output port for the frame, the output and input port can be verified to be in the correct VLAN domain if VLAN verification is enabled for the port and the frame contains a VLAN tag. The VLAN resolution table (see VLAN Resolution Table) is used as follows:

- If the frame's VLAN ID is in the table but the output port number is not a member of the VLAN domain, or the input port is not a member of the VLAN domain, the frame is marked invalid and is eventually discarded.
- If the frame's VLAN ID is in the table and the output and input ports are members of the VLAN domain, the frame is forwarded normally.
- If the frame's VLAN ID is not found in the VLAN table or the frame has no VLAN tag, the frame is forwarded normally (default broadcast domain), or if the discard bit for the port is set (also in register ESW_VLANV) it is discarded.

43.5.9.2 Broadcast/Multicast/VLAN Domain Resolution

To ensure that traffic within VLAN channels are always routed to the correct ports, for example to avoid the duplication of critical information through a network, the switch implements a resolution mechanism that, for any frame that is switched to multiple ports, checks the VLAN ID provided with the current frame.

The VLAN resolution mechanism searches the VLAN resolution table (see ESW_VRES*n* registers), which stores up to 32 unique VLAN IDs, each associated to a port bit mask. The resolution mechanism is used for the following conditions:

- Unicast frames with a destination MAC address that are not in the table of the layer 2 engine
- Multicast frames with a destination MAC address that are not in the table of the layer 2 engine
- Any broadcast frame

43.5.9.2.1 VLAN Resolution Table

The VLAN resolution table (ESW_VRES*n*) provides a unique VLAN ID/port bit mask association for up to 32 VLANs. A default entry (ESW_DBCR) provides an additional port bit mask. The port bit mask implements one bit for each port.

Each port bit indicates, if set, that it is member of the VLAN and frames with the corresponding VLAN ID can be switched to the port. If the port bit is cleared, a frame with the corresponding VLAN ID is not switched to that port. If no VLAN ID matches, the default mask is applied.

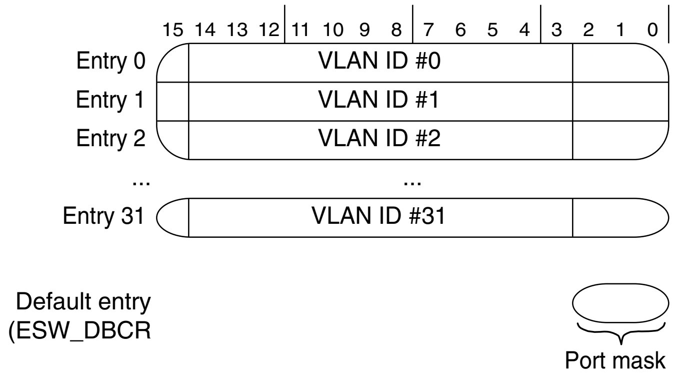


Figure 43-130. VLAN Resolution Table Overview

43.5.9.2.2 VLAN Switching / Resolution Mechanism

The VLAN table is used for VLAN domain verification (see VLAN Domain Verification) and VLAN resolution. Once the frame has passed any VLAN domain verification (i.e. will not be discarded by the verification function already) the forwarding resolution applies.

- If the destination MAC address (Unicast or Multicast) is found in the MAC address table and
  - the frame carries a VLAN tag that is found in the VLAN table, the frame can be forwarded only to the ports within the VLAN domain and will be discarded if the destination port is not member of the VLAN domain.
  - else if the frame carries a VLAN tag that is not found in the VLAN table, or does not contain a VLAN tag, it is forwarded as indicated by the lookup table (note that VLAN domain verification can be configured to discard the frame in this case if enabled).
- If the destination MAC address (Unicast or Multicast) is not found in the MAC address table, or if the destination address is the Broadcast address, the frame is forwarded according to the following rules:
  - If the frame carries a VLAN tag, the VLAN resolution table is searched for a matching VLAN ID and the frame is sent to all ports that are associated with the VLAN ID.
  - If the frame carries a VLAN tag and the VLAN ID does not match any entry in the VLAN Resolution Table, the frame is forwarded to all ports that are enabled to receive broadcast frames by the default entry.

- If the frame does not carry a VLAN tag the frame is forwarded to all ports that are enabled to receive broadcast frames by the default entry.
- The frame is discarded, if it cannot be associated with any VLAN group and if the default (broadcast) group has been set to all zero.

To disable the VLAN resolution set all VLAN IDs to 0xFFF or (0x000 if that ID is not used) and all port mask bits to 1. If the VLAN resolution is disabled, normal port flooding is implemented as described in Frame Classification and Priority Resolution. The default entry can still be used to restrict broadcast to only dedicated ports, if not programmed to all 1s.

43.5.9.3 Port Mirroring

The function allows duplicating traffic to a dedicated mirror port. Any one of the ports can be assigned to act as a mirror port (register ESW_MCR).

The mirror port then is always added to the list of output ports and therefore receives a copy of the frame, if any of the following rules matches with the currently processed frame:

- Ingress Port Number Match

  When a frame is received on port N and the corresponding bit in the register ESW_INGMAP is set to 1, the frame is mirrored.
- Egress Port Number Match

  When a frame is forwarded to port N and the corresponding bit in the register ESW_EGMAP is set to 1, the frame will be mirrored.
- MAC Ingress SA Match

  When the Ingress Port Number match succeeded (see above) and the MAC source address matches the ESW_INGSA{L,H}, the frame will be mirrored.
- MAC Ingress DA Match

  When the Ingress Port Number match succeeded (see above) and the MAC destination address matches the ESW_INGDA{L,H}, the frame will be mirrored.
- MAC Egress SA Match

  When the Egress Port Number match succeeded (see above) and the MAC source address matches the ESW_EGSA{L,H}, the frame will be mirrored.
- MAC Egress DA Match

When the Egress Port Number match succeeded (see above) and the MAC destination address matches the ESW_EGDA{L,H}, the frame will be mirrored.

In addition, a counter is implemented (Register MIRROR_COUNT) that allows specifying that only every Nth frame that matches any of the above criteria is mirrored. If the counter is set to 1 or 0, every frame is mirrored that matches any of the above criteria.

43.5.9.4 Protocol Snooping

The incoming frames are parsed for IPv4 and IPv6 headers and UDP/TCP if available. The snooping function can be programmed to redirect specific protocols exclusively to the management port. See IP Snooping and TCP/UDP Port Number Snooping for a description of the snooping options.

The snooping is active only for frames received from the external ports. When a frame is transmitted from the management port itself, snooping does not apply and the frames are forwarded normally (MAC lookup).

43.5.9.5 Bridge Protocol Frame Resolution

To implement bridge control protocols like the Spanning Tree protocol, the following control functions are performed by the Protocol Frame Resolution function:

43.5.9.5.1 Input Port Blocking

The input port blocking function is used to avoid forwarding of frames after address learning. The firmware can program the ESW_BKLR register and if a frame is received on port *n* that should be blocked (BE*n* = 1) and the frame is not a bridge protocol frame (see below), the frame is marked for discard and is not forwarded to any output port.

43.5.9.5.2 Input Port Learning Disable

To reduce processing load from the firmware, a port can be configured for exclusion from learning (see the ESW_BKLR register).

When learning is disabled on a port no source address extraction happens for incoming frames, with the exception of incoming BPDU frames. BPDU frame source addresses are always extracted and forwarded to the learning interface.

43.5.9.5.3 Management Port Forwarding

If enabled, bridge protocol frames are always forwarded to the dedicated management port (see the ESW_BMPC register) independent of any address lookup or other resolution functions.

Bridge protocol frames are identified by its destination address being any of the following:

- 01-80-C2-00-00-00 to 01-80-C2-00-00-0F (Spanning Tree, IEEE 802.1d, Table 7-9)
- 01-80-C2-00-00-10 (Bridge Management Address, 802.1d, Table 7-10)
- 01-80-C2-00-00-20 to 01-80-C2-00-00-2F (Generic Attribute Registration Protocol, 802.1d, Table 12-1)

43.5.9.5.4 Management Frame Forwarding

If the management port transmits Frames, they are forwarded according to the port mask defined in the configuration register ESW_BMPC. A handshaking mechanism is implemented that can be used by the firmware to configure the destination port mask on a frame-by-frame basis for management frames.

Note: VLAN domain verification/discard (see the ESW_VLANV register) should be switched off for the management port to avoid that the switch discards management frames.

43.5.9.6 Congestion Resolution

The congestion resolution function is used whenever an output port is not available and data needs to be sent to that port. An output port is defined to be available if the port is enabled (bit in ESW_PER set 1) and the output buffer (shared memory) is not congested. If, for a port, one of these conditions is not valid, the port is not available and frames cannot be switched to that port.

The congestion resolution function determines whether the frame should be processed further or discarded according to the following rules:

43.5.9.6.1 Unique Destination (one input to one output)

If the output port is enabled and can accept a frame the frame will be forwarded normally.

In any other case the frame will be discarded. If a frame switched to port N, the counter ESW_P*n*OQC is incremented.

43.5.9.6.2 Multiple Destinations (Flooding)

After broadcast / flooding resolution a frame needs to be switched to multiple output ports.

- Output disabled: All disabled ports are removed from the list of outputs.
- Output congestion: If any of the outputs cannot accept a frame (As indicated by the output queue management for the port, implementation specific) it is also removed from the list of outputs.

If no output port is left in the list of outputs, the frame is discarded.

If a frame switched to port N, the counter ESW_P*n*OQC is incremented.

43.5.9.7 Switching

After the output port(s) have been determined, the switch control enables the corresponding path though the Switch Matrix and the frame is forwarded to the output queue(s).

In a similar fashion, if a Frame should be switched to multiple ports (e.g. Broadcast), the switch control enables the corresponding paths though the Switch Matrix and the frame is forwarded to all the destination output ports.

43.5.10 Output Frame Queuing

The memory controller implements a shared memory architecture to store Frames of arbitrary size for multiple destination ports.

Each destination port implements 4 priority queues. The memory controller implements a single write input port and three output ports with the capability to perform virtual frame duplication on the output ports (Multiple reads on multiple ports of a single frame stored in the buffer).

A single large memory, partitioned in 256 byte cells, is implemented to efficiently share the available space for small and large frames without leaving large unused spaces when storing small frames.

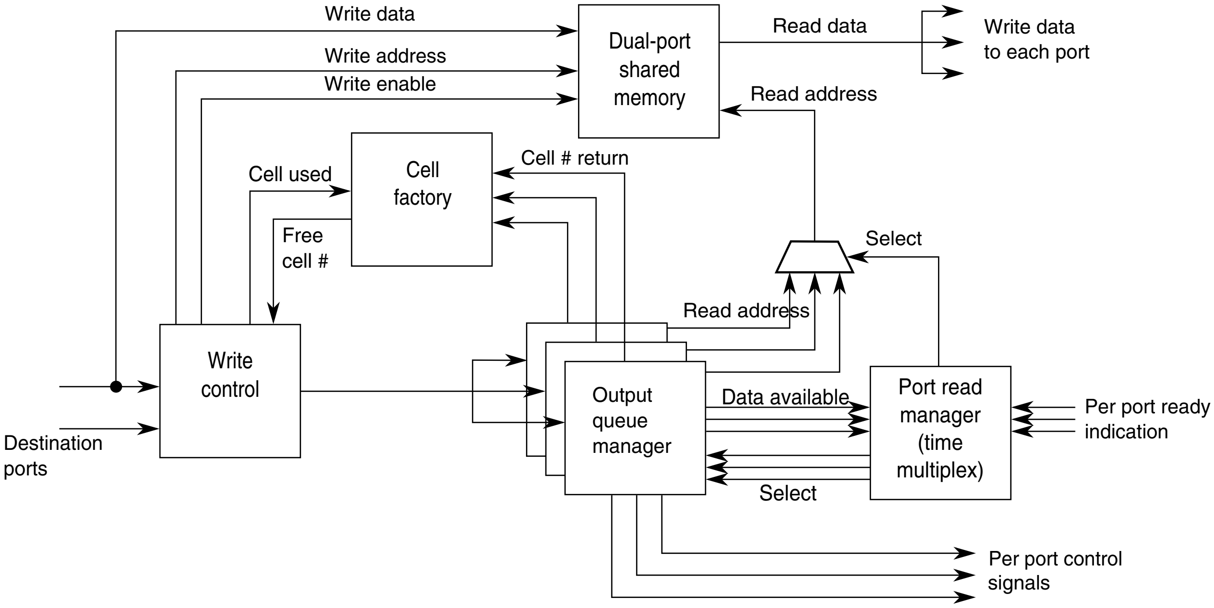


Figure 43-131. Memory Controller Overview

43.5.10.1 Cell and Queue Concept

The shared memory is partitioned in 256 byte cells using a 32-bit datapath implementation. This results in a cell holding 64 32-bit words.

Incoming frames are stored, partitioned in cells, in the shared memory and only the cell numbers are managed by the individual port queues.

Due to the arbitrary length of incoming frames, the last cell may not be fully utilized. A frame can spread from one to any number of cells. The number of bytes used in the last cell is also stored in the queue FIFO together with the individual cell numbers.

Cells can be stored anywhere in the shared memory. A single frame must not necessarily be stored in consecutive cells but instead can be scattered over the complete memory at arbitrary positions. The start of a cell is fixed to a 64-word boundary (i.e. the memory start address of a cell is simply the cell # multiplied by 64).

Per port, a queue FIFO is implemented that stores the cell numbers for the frames The number of bytes used in the last cell is also stored in the queue FIFO together with the individual cell numbers.

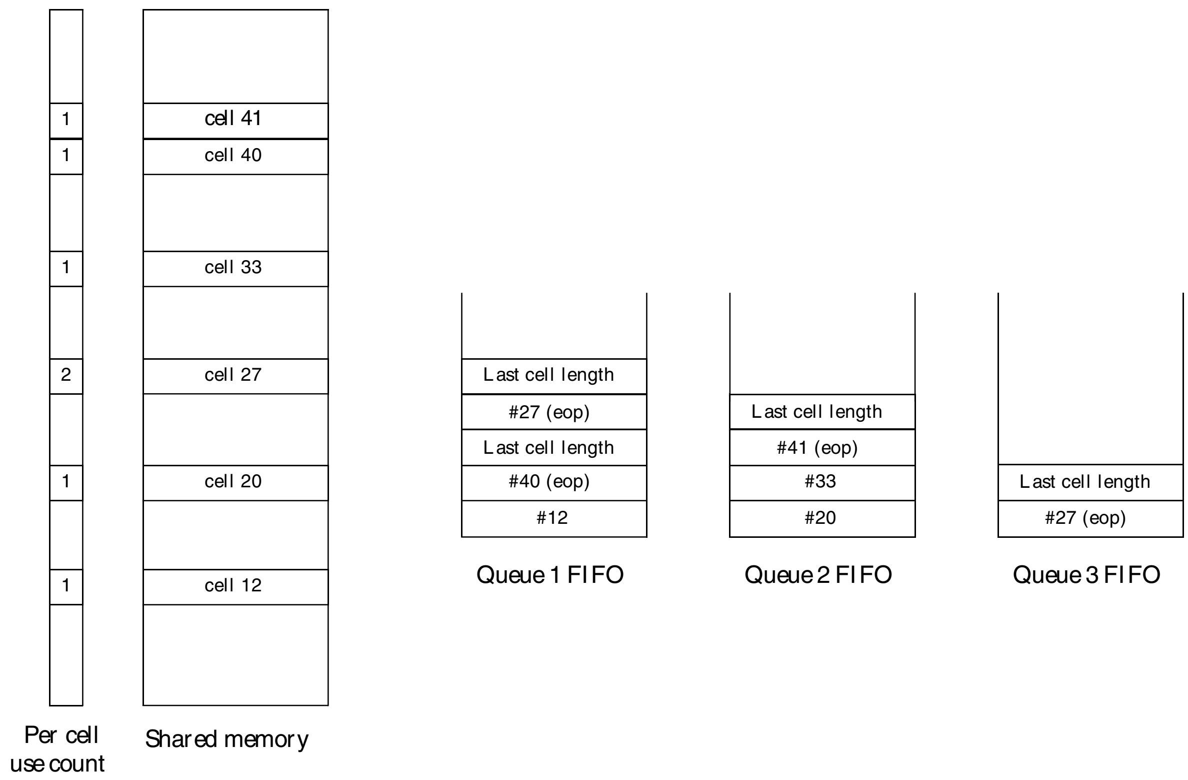


Figure 43-132. Cell storage concept

The example in the figure above shows the storage of three frames with one frame duplicated and stored in two queues.

43.5.10.2 Write Control Module

The Write Control Module receives frames from the Switch engine, partitions and stores the frames in the shared memory. The cell numbers used to store the frame are forwarded to the port output managers.

43.5.10.3 Cell Factory Module

The Cell Factory implements the cell management, it always provides a free cell number to the write control module so the write control module can immediately start writing into memory to avoid any write latency.

43.5.10.4 Output Queue Manager

The Output Queue Manager implements, per port, the individual queue FIFOs (One queue FIFO per priority). The queue FIFOs are addressed by the priority information extracted by the Switch classification engine. Per port, eight prioritized queues are implemented.

When more than one queue has data available, the read logic selects one of the queues with a Weighted Fair Queuing scheduling algorithm.

43.5.10.4.1 Weighted Fair Queuing Scheduling Algorithm

The weight of each queue, common for all ports, can be configured between 0 and 30 with the register ESW_QWT. A Queue with a higher weight is served more often than a queue with lower weight.

Queue 0 represents the lowest priority, Queue 3 the highest priority queue.

The scheduler first serves the queue with the highest weight. Each time the scheduler serves a queue, the weight of the other queues is increased and the weight of the selected queue is reset (decreased) to its programmed weight value. This guarantees that all queues are served eventually.

When multiple queues have the same weight the queue with the higher number is served first.

If all weights are programmed to 0 (default) a strict priority scheme is active where the higher priority queues are served as long as they are not empty.

43.5.10.5 Congestion Management

The Write control logic is protected against memory overflow. When data is written is the cell factory has no more free cells (Number of available cells less than value programmed in register ESW_LMT), the frame is discarded or terminated with an error (i.e. forwarded to the output queue manager with the end-of-packet and error indication).

If the congestion persists, the switch resolves the congestion as specifies in Congestion Resolution.

43.5.11 Reset and stop functions

43.5.11.1 Stop Controls

The register ESW_MODE offers several bits that control output pins. In addition some controls have an effect on internal logic functions:

- stop_en: no internal function.
- switch_en: when de-asserted, all DMA registers are cleared.
- switch_reset: no internal function.

An external logic may use the controls to disable or enable the switch function as necessary.

43.5.11.2 Port Disable

The switch toplevel offers a disable input for each port (port_dis(2:0). When a pin is asserted (1), the corresponding port enable bits within register ESW_PER for both transmit and receive will be cleared.

This results in the following behavior:

- If the port-enable bits of port0 are cleared, it also resets the input buffer and output buffer at the port0 DMA interface.
  - The ready output to DMA0 will be asserted allowing the application to continue writing data at the interface, which will be ignored (application flush). No further transmitted frame status will be given (i.e. for any currently stored if any, as well as the currently ignored).
  - If the transmit enable is cleared while the interface is currently transferring a frame to the DMA, the frame is aborted (output buffer reset). The eop is not produced. Therefore the connected DMA module must be reset to ensure proper restart after re-enabling the port.
- If any port's transmit enable bit is cleared, the shared memory will continue delivering the frames stored currently for a port as normal (i.e. flushing the memory). New frames will be discarded before they are written into the shared memory. That is, no invalid frame will appear on the MAC interfaces after disabling or re-enabling a port.
- If any port's receive enable bit is cleared while a frame is transferred, this frame will be aborted with an error internally. The port's ready indication will stay asserted to flush any application data. Reenabling the port at any time will ignore any input data until a sop starts a new frame.

43.5.11.3 Port 0 Input Protection

The port 0 input buffer is protected for application errors that abort a frame without writing a proper eop to the interface. The next frame then written to the port 0 transmit interface will be concatenated with whatever data was already written before, but the frame will be marked with an error and hence will be forwarded and transmitted with an error indication (mii tx error).

If the port0 input buffer is reset (by deasserting ESW_PER receive enable bit) while a frame is transferred to the switch internally, the frame transfer will be aborted in a clean way (producing an eop with error indication) to avoid blocking the switch.

43.5.11.4 Port 1/2 Input Protection

Ports 1 and 2 are protected for a 2nd SOP in case the MAC is reset in the middle of a receive transaction to the switch hence did not produce a proper EOP to the switch. The next frame will be concatenated with whatever was provided to the switch before and marked with an error.

If the MAC is stopped in the middle of a transaction, the switch is blocked, waiting for the EOP, not serving any of the other ports. Clearing the ESW_PER receive enable bit in this situation will terminate the frame with an error internally to the switch hence remove the blocking condition.

43.5.11.5 DMA Bus Error

When the DMA bus error input (dma_eberr_int) is asserted, the DMA registers are all cleared. If the corresponding interrupt was enabled the ipi_eberr_int pin will be asserted.

Chapter 44
Universal Serial Bus (USB) Controller

44.1 Overview

The USB controller block provides high performance USB functionality that complies with the USB 2.0 specification. It is a dual role device which can be configured to act as Host or Device through firmware interface. It is not a true OTG controller.

The USB controller consists of two independent dual role(host/device) USB controller cores. Each controller core has a UTMI interface and is connected to on-chip PHY. For a detailed description about the features, please refer to Features. It is used as both a downstream and upstream port.

The following figure is a block diagram of USB.

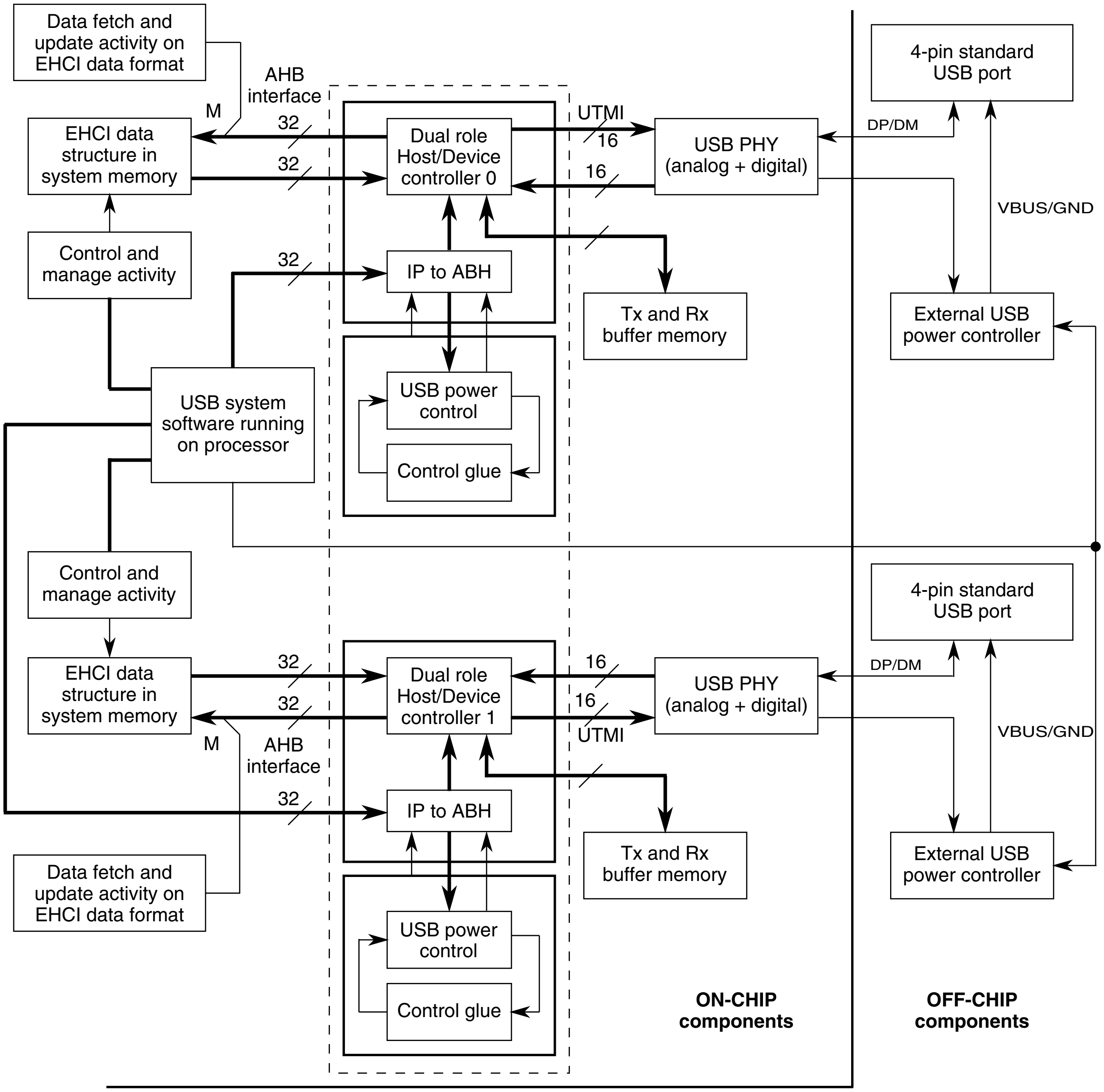


Figure 44-1. USB block diagram

44.2 Features

The USB includes the following features:

- High-speed dual role Host/device controller core
  - HS/FS/LS UTMI compliant interface
  - High Speed, Full Speed and Low Speed operation in Host mode
  - High Speed, and Full Speed operation in Peripheral mode
  - Up to six bidirectional software configurable endpoints

- Low-power mode with local and remote wake-up capability
- Serial PHY interfaces configurable for bidirectional/unidirectional and differential/single ended
- Embedded DMA controller.

44.2.1 Ports

These are detailed list of Ports on each USB devices on Vybrid:

- USB_DCAP — Common for both of the controller. Connected are two 1.1 pf cap on them.
- USB0_DM — Connect directly to USB four pin connectors DM pin.
- USB0_DP — Connect directly to USB four pin connectors DP pin.
- USB0_GND — Connect directly to USB four pin connectors GND pin also, connect it to ground of board.
- USB0_VBUS — Connect to external power controller.

If USB is configured as:

- Self powered device — The external power controller will drive it with 5 V supply.
- Bus powered device — The external power controller will connect it to VBUS of cable.
- Host, connected to bus powered low power device then external power controller will drive 100 mA of current to VBUS of cable.
- Host, connected to bus powered high power device then external power controller will drive 500 mA of current to VBUS of cable.

If the consumed current is exceeding the rated current then the external power controller will report Vybrid on USBx_VBUS_OC and in that case, Vybrid USB will suspend the port.

- USB0_VBUS_DETECT — Connected to cable VBUS in case of any mode. External power controller need to drive this on detection of cable connected.
- USB0_VBUS_EN — Will be connected to external power controller. This indicates to enable the VBUS supply.
- USB0_VBUS_OC — Indicated by external power controller about connected device is fetching more current than the configured one.
- USB0_SOF_PULSE — Can be used to configure sample rate to any of external audio/video sampler to match rate with current USB bus speed.

44.2.1.1 Modes of Operation

The USB has two main modes of operation: normal mode and low power mode. Each USB controller core can be configured for High Speed operation (480 Mbps), Full Speed operation (12Mbps) and Low Speed operation (1.5 Mbps).

This chapter explains the operation modes.

44.2.1.2 Normal Mode

The USB controller can be configured to work in either Host mode or Device (Peripheral mode).

The USB is not a true OTG. It can be configured by software to function either as peripheral or as host. The ID pin, which is unique for OTG operation, is not present in this implementation. There are no five pin interface. The user will get four pin host/ device interface.

44.2.1.3 Low Power Mode

Each USB PHY has a low power mode (Low power Suspend Mode) to reduce power consumption. It can be entered by setting bit "PHCD" in USB_PORTSC1 register. This bit must only be set after the bus is in suspend state. When operating as host, software can set the SUSP bit in the PORTC1 register to suspend the bus. When operating in peripheral mode, the controller will automatically enter suspend mode when the bus has been idle for 3 ms. At that time, software can enable the low-power suspend feature in the PHY.

Either the local ARM platform or the remote USB Host/Peripheral can initiate a wake-up sequence to resume USB communication. For detail about Suspend/Resume, please see USB Power Control Block.

44.3 Non-core Registers

There are two kinds of registers in the USB module: USB core registers and USB non-core registers.

USB core registers are used to control USB core functions, and more independent of USB features. Each USB controller core has its own core registers.

USB non-core registers are additional to USB core registers. These registers are implementation specific. The registers and the bit mapping may vary across different devices.

This section describes only the USB non-core registers. For detailed descriptions of USB core registers, please refer toRegister Interface.

NOTE

For reserved bits, please preserve the value when writing (read its reset value, then write this value back).

"USB" prefix in register name indicates it is a core register for USB controller core.

"USBC" prefix in register name indicates it is a USB non-core register.

USBC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_5800	USB Control Register (USBC0_CTRL)	32	R/W	0000_3000h	44.3.1/2144
4003_5818	UTMI PHY Control Register (USBC0_PHY)	32	R/W	0000_0000h	44.3.2/2147
400B_5800	USB Control Register (USBC1_CTRL)	32	R/W	0000_3000h	44.3.1/2144
400B_5818	UTMI PHY Control Register (USBC1_PHY)	32	R/W	0000_0000h	44.3.2/2147

44.3.1 USB Control Register (USBCx_CTRL)

The USB control register controls features that are not directly related to the USB functionality, but control special features, interfacing on the USB ports, as well as power control and wake-up functionality.

Address: Base address + 800h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WIR	Reserved					VBUS_CH_INT_MASK	VBUS_CH_STAT	Reserved						WKUP_VBUS_EN	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WKUP_SW	WKUP_SW_EN	UTMI_ON_CLOCK	SUSPENDM	RESET	WIE	PM	OVER_CUR_POL	OVER_CUR_DIS	Reserved						
W																
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

USBCx_CTRL field descriptions

Field	Description
31 WIR	USB Wake-up Interrupt Request This bit indicates that a wake-up interrupt request is received on the USB port. This bit is cleared by disabling the wake-up interrupt (clearing bit "WIE"). 1 Wake-up Interrupt Request received 0 No wake-up interrupt request received

Table continues on the next page...

USBCx_CTRL field descriptions (continued)

Field	Description
30–26 Reserved	This field is reserved.
25 VBUS_CH_INT_ MASK	When this bit is set, generation of VBUS_EN_INTR is masked. 0 VBUS_EN_INTR not masked 1 VBUS_EN_INTR masked
24 VBUS_CH_STAT	This bit is set by host controller whenever Port Power Enable/Disable event occurs. 0 No Change in Port Power 1 Change in Port Power Control
23–18 Reserved	This field is reserved.
17 WKUP_VBUS_ EN	Wake-up on VBUS change enable 1 Enable 0 Disable
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 WKUP_SW	USB Core Software Wake-up 1 Force wake-up 0 Inactive
14 WKUP_SW_EN	USB Core Software Wake-up Enable 1 Enable 0 Disable
13 UTMI_ON_ CLOCK	Force USB Core UTMI PHY clock output on even if suspend mode. 1 Force clock output on 0 Inactive
12 SUSPENDM	Force OTG UTMI PHY Suspend This bit is used to force PHY into low-power mode. During normal operation, software should set bit PHCD, after the bus is suspended, in USB core register PORTSC1 to put PHY into low-power mode. For Freescale test only. 1 Inactive 0 Force OTG UTMI PHY Suspend
11 RESET	Force USB UTMI PHY Reset Not for use in normal operation. During normal operation, software should set USBCMD. RST bit to reset the UTMI PHY 1 Reset the PHY 0 Inactive
10 WIE	Wake-up Interrupt Enable This bit enables or disables the wake-up interrupt. Disabling the interrupt also clears the Interrupt request bit. Wake-up interrupt enable should be turned off after receiving a wake-up interrupt and turned on again prior to going in suspend mode

Table continues on the next page...

USBCx_CTRL field descriptions (continued)

Field	Description
	1 Interrupt Enabled 0 Interrupt Disabled
9 PM	USB Power Mask 1 The USBPWR and OC pins are not used by the USB core. 0 The USBPWR pin will assert with the USB core's Vbus power Enable and the assertion of the OC input will be reported to the USB core.
8 OVER_CUR_ POL	USB Polarity of Overcurrent 1 Low active 0 High active
7 OVER_CUR_DIS	Disable USB Overcurrent Detection 1 Disables overcurrent detection 0 Enables overcurrent detection
6–0 Reserved	This field is reserved. Reserved

44.3.2 UTMI PHY Control Register (USBCx_PHY)

Address: Base address + 818h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UTMI_CLK_VLD	0														
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													CHRGDET	CHRGDET_INT_EN	CHRGDET_INT_FLG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBCx_PHY field descriptions

Field	Description
31 UTMI_CLK_VLD	UTMI PHY Clock Valid. This bit indicates that UTMI clock is valid. 0 UTMI Clock is invalid 1 UTMI Clock is valid
30–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CHRGDET	UTMI PHY chrgdet. Charger detector output. Actual circuit is not present. It is a self powered USB, we do not need charger circuitry but these bits are present to allow software compatibility.

Table continues on the next page...

USBCx_PHY field descriptions (continued)

Field	Description
	NOTE: Maximum response time = 1us 0 When a Host is detected 1 When a charger is detected
1 CHRGDET_INT_ EN	Charger detected interrupt enable. Actual circuit is not present. It is a self powered USB, we do not need charger circuitry but these bits are present to allow software compatibility. 0 Disable 1 Enable
0 CHRGDET_INT_ FLG	Charger detected interrupt flag. This bit is cleared when CHRGDET_INT_EN is 0. Actual circuit is not present. It is a self powered USB, we do not need charger circuitry but these bits are present to allow software compatibility. 0 No charger detected interrupt 1 Charger detected interrupt occurred

44.4 Core Registers

USB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_4000	Identification register (USB0_ID)	32	R	E481_FA05h	44.4.1/2152
4003_4004	Hardware General (USB0_HWGENERAL)	32	R	See section	44.4.2/2153
4003_4008	Host Hardware Parameters (USB0_HWHOST)	32	R	1002_0001h	44.4.3/2154
4003_400C	Device Hardware Parameters (USB0_HWDEVICE)	32	R	0000_000Dh	44.4.4/2154
4003_4010	TX Buffer Hardware Parameters (USB0_HWTXBUF)	32	R	8008_0B04h	44.4.5/2155
4003_4014	RX Buffer Hardware Parameters (USB0_HWRXBUF)	32	R	0000_0904h	44.4.6/2155
4003_4080	General Purpose Timer #0 Load (USB0_GPTIMER0LD)	32	R/W	0000_0000h	44.4.7/2156
4003_4084	General Purpose Timer #0 Controller (USB0_GPTIMER0CTRL)	32	R/W	0000_0000h	44.4.8/2157
4003_4088	General Purpose Timer #1 Load (USB0_GPTIMER1LD)	32	R/W	0000_0000h	44.4.9/2158
4003_408C	General Purpose Timer #1 Controller (USB0_GPTIMER1CTRL)	32	R/W	0000_0000h	44.4.10/ 2158
4003_4090	System Bus Config (USB0_SBUSCFG)	32	R/W	0000_0001h	44.4.11/ 2160
4003_4100	Capability Register Length (USB0_CAPLENGTH)	8	R	40h	44.4.12/ 2160
4003_4102	Host Controller Interface Version (USB0_HCIVERSION)	16	R	0100h	44.4.13/ 2161

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_4104	Host Controller Structural Parameters (USB0_HCSPARAMS)	32	R	0001_0011h	44.4.14/ 2161
4003_4108	Host Controller Capability Parameters (USB0_HCCPARAMS)	32	R	0000_0006h	44.4.15/ 2163
4003_4120	Device Controller Interface Version (USB0_DCIVERSION)	16	R	0001h	44.4.16/ 2164
4003_4124	Device Controller Capability Parameters (USB0_DCCPARAMS)	32	R	0000_0186h	44.4.17/ 2164
4003_4140	USB Command Register (USB0_USBCMD)	32	R/W	0008_0000h	44.4.18/ 2165
4003_4144	USB Status Register (USB0_USBSTS)	32	R/W	0000_0000h	44.4.19/ 2169
4003_4148	Interrupt Enable Register (USB0_USBINTR)	32	R/W	0000_0000h	44.4.20/ 2173
4003_414C	USB Frame Index (USB0_FRINDEX)	32	R/W	0000_0000h	44.4.21/ 2175
4003_4154	Frame List Base Address (USB0_PERIODICLISTBASE)	32	R/W	0000_0000h	44.4.22/ 2176
4003_4154	Device Address (USB0_DEVICEADDR)	32	R/W	0000_0000h	44.4.23/ 2177
4003_4158	Next Asynch. Address (USB0_ASYNCLISTADDR)	32	R/W	0000_0000h	44.4.24/ 2178
4003_4158	Endpoint List Address (USB0_ENDPTLISTADDR)	32	R/W	0000_0000h	44.4.25/ 2178
4003_4160	Programmable Burst Size (USB0_BURSTSIZE)	32	R/W	0000_1010h	44.4.26/ 2179
4003_4164	TX FIFO Fill Tuning (USB0_TXFILLTUNING)	32	R/W	0000_0808h	44.4.27/ 2179
4003_4178	Endpoint NAK (USB0_ENDPTNAK)	32	R/W	0000_0000h	44.4.28/ 2181
4003_417C	Endpoint NAK Enable (USB0_ENDPTNAKEN)	32	R/W	0000_0000h	44.4.29/ 2181
4003_4184	Port Status & Control (USB0_PORTSC1)	32	R/W	1000_0000h	44.4.30/ 2182
4003_41A4	On-The-Go Status & control (USB0_OTGSC)	32	R/W	0000_0220h	44.4.31/ 2188
4003_41A8	USB Device Mode (USB0_USBMODE)	32	R/W	0000_5002h	44.4.32/ 2191
4003_41AC	Endpoint Setup Status (USB0_ENDPTSETUPSTAT)	32	R/W	0000_0000h	44.4.33/ 2192
4003_41B0	Endpoint Initialization (USB0_ENDPTPRIME)	32	R/W	0000_0000h	44.4.34/ 2193
4003_41B4	Endpoint De-Initialize (USB0_ENDPTFLUSH)	32	R/W	0000_0000h	44.4.35/ 2194

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_41B8	Endpoint Status (USB0_ENDPTSTAT)	32	R	0000_0000h	44.4.36/ 2194
4003_41BC	Endpoint Complete (USB0_ENDPTCOMPLETE)	32	R/W	0000_0000h	44.4.37/ 2195
4003_41C0	Endpoint Control0 (USB0_ENDPTCTRL0)	32	R/W	0080_0080h	44.4.38/ 2196
4003_41C4	Endpoint Controln (USB0_ENDPTCTRL1)	32	R/W	0000_0000h	44.4.39/ 2197
4003_41C8	Endpoint Controln (USB0_ENDPTCTRL2)	32	R/W	0000_0000h	44.4.39/ 2197
4003_41CC	Endpoint Controln (USB0_ENDPTCTRL3)	32	R/W	0000_0000h	44.4.39/ 2197
4003_41D0	Endpoint Controln (USB0_ENDPTCTRL4)	32	R/W	0000_0000h	44.4.39/ 2197
4003_41D4	Endpoint Controln (USB0_ENDPTCTRL5)	32	R/W	0000_0000h	44.4.39/ 2197
400B_4000	Identification register (USB1_ID)	32	R	E481_FA05h	44.4.1/2152
400B_4004	Hardware General (USB1_HWGENERAL)	32	R	See section	44.4.2/2153
400B_4008	Host Hardware Parameters (USB1_HWHOST)	32	R	1002_0001h	44.4.3/2154
400B_400C	Device Hardware Parameters (USB1_HWDEVICE)	32	R	0000_000Dh	44.4.4/2154
400B_4010	TX Buffer Hardware Parameters (USB1_HWTXBUF)	32	R	8008_0B04h	44.4.5/2155
400B_4014	RX Buffer Hardware Parameters (USB1_HWRXBUF)	32	R	0000_0904h	44.4.6/2155
400B_4080	General Purpose Timer #0 Load (USB1_GPTIMER0LD)	32	R/W	0000_0000h	44.4.7/2156
400B_4084	General Purpose Timer #0 Controller (USB1_GPTIMER0CTRL)	32	R/W	0000_0000h	44.4.8/2157
400B_4088	General Purpose Timer #1 Load (USB1_GPTIMER1LD)	32	R/W	0000_0000h	44.4.9/2158
400B_408C	General Purpose Timer #1 Controller (USB1_GPTIMER1CTRL)	32	R/W	0000_0000h	44.4.10/ 2158
400B_4090	System Bus Config (USB1_SBUSCFG)	32	R/W	0000_0001h	44.4.11/ 2160
400B_4100	Capability Register Length (USB1_CAPLENGTH)	8	R	40h	44.4.12/ 2160
400B_4102	Host Controller Interface Version (USB1_HCIVERSION)	16	R	0100h	44.4.13/ 2161
400B_4104	Host Controller Structural Parameters (USB1_HCSPARAMS)	32	R	0001_0011h	44.4.14/ 2161
400B_4108	Host Controller Capability Parameters (USB1_HCCPARAMS)	32	R	0000_0006h	44.4.15/ 2163
400B_4120	Device Controller Interface Version (USB1_DCIVERSION)	16	R	0001h	44.4.16/ 2164
400B_4124	Device Controller Capability Parameters (USB1_DCCPARAMS)	32	R	0000_0186h	44.4.17/ 2164

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_4140	USB Command Register (USB1_USBCMD)	32	R/W	0008_0000h	44.4.18/ 2165
400B_4144	USB Status Register (USB1_USBSTS)	32	R/W	0000_0000h	44.4.19/ 2169
400B_4148	Interrupt Enable Register (USB1_USBINTR)	32	R/W	0000_0000h	44.4.20/ 2173
400B_414C	USB Frame Index (USB1_FRINDEX)	32	R/W	0000_0000h	44.4.21/ 2175
400B_4154	Frame List Base Address (USB1_PERIODICLISTBASE)	32	R/W	0000_0000h	44.4.22/ 2176
400B_4154	Device Address (USB1_DEVICEADDR)	32	R/W	0000_0000h	44.4.23/ 2177
400B_4158	Next Asynch. Address (USB1_ASYNCLISTADDR)	32	R/W	0000_0000h	44.4.24/ 2178
400B_4158	Endpoint List Address (USB1_ENDPTLISTADDR)	32	R/W	0000_0000h	44.4.25/ 2178
400B_4160	Programmable Burst Size (USB1_BURSTSIZE)	32	R/W	0000_1010h	44.4.26/ 2179
400B_4164	TX FIFO Fill Tuning (USB1_TXFILLTUNING)	32	R/W	0000_0808h	44.4.27/ 2179
400B_4178	Endpoint NAK (USB1_ENDPTNAK)	32	R/W	0000_0000h	44.4.28/ 2181
400B_417C	Endpoint NAK Enable (USB1_ENDPTNAKEN)	32	R/W	0000_0000h	44.4.29/ 2181
400B_4184	Port Status & Control (USB1_PORTSC1)	32	R/W	1000_0000h	44.4.30/ 2182
400B_41A4	On-The-Go Status & control (USB1_OTGSC)	32	R/W	0000_0220h	44.4.31/ 2188
400B_41A8	USB Device Mode (USB1_USBMODE)	32	R/W	0000_5002h	44.4.32/ 2191
400B_41AC	Endpoint Setup Status (USB1_ENDPTSETUPSTAT)	32	R/W	0000_0000h	44.4.33/ 2192
400B_41B0	Endpoint Initialization (USB1_ENDPTPRIME)	32	R/W	0000_0000h	44.4.34/ 2193
400B_41B4	Endpoint De-Initialize (USB1_ENDPTFLUSH)	32	R/W	0000_0000h	44.4.35/ 2194
400B_41B8	Endpoint Status (USB1_ENDPTSTAT)	32	R	0000_0000h	44.4.36/ 2194
400B_41BC	Endpoint Complete (USB1_ENDPTCOMPLETE)	32	R/W	0000_0000h	44.4.37/ 2195
400B_41C0	Endpoint Control0 (USB1_ENDPTCTRL0)	32	R/W	0080_0080h	44.4.38/ 2196
400B_41C4	Endpoint Controln (USB1_ENDPTCTRL1)	32	R/W	0000_0000h	44.4.39/ 2197

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_41C8	Endpoint Controln (USB1_ENDPTCTRL2)	32	R/W	0000_0000h	44.4.39/ 2197
400B_41CC	Endpoint Controln (USB1_ENDPTCTRL3)	32	R/W	0000_0000h	44.4.39/ 2197
400B_41D0	Endpoint Controln (USB1_ENDPTCTRL4)	32	R/W	0000_0000h	44.4.39/ 2197
400B_41D4	Endpoint Controln (USB1_ENDPTCTRL5)	32	R/W	0000_0000h	44.4.39/ 2197

44.4.1 Identification register (USBx_ID)

The ID register identifies the USB 2.0 OTG High-Speed core and its revision.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								REVISION							
W																
Reset	1	1	1	0	0	1	0	0	1	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		NID						Reserved		ID					
W																
Reset	1	1	1	1	1	0	1	0	0	0	0	0	0	1	0	1

USBx_ID field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 REVISION	Revision number of the controller core.
15–14 -	This field is reserved. Reserved
13–8 NID	Complement version of ID
7–6 -	This field is reserved. Reserved
5–0 ID	Configuration number. This number is set to 0x05 and indicates that the peripheral is USB 2.0 OTG High-Speed core.

44.4.2 Hardware General (USBx_HWGENERAL)

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					SM		PHYM			PHYW		Reserved			
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	1*	0*	1*	0*	1*

* Notes:
- The reset value depends on the implementation.

USBx_HWGENERAL field descriptions

Field	Description
31–11 -	This field is reserved. Reserved
10–9 SM	Serial interface mode capability For OTG controller core, reset value is '00b'. 00 No Serial Engine, always use parallel signaling. 01 Serial Engine present, always use serial signaling for FS/LS. 10 Software programmable - Reset to use parallel signaling for FS/LS 11 Software programmable - Reset to use serial signaling for FS/LS
8–6 PHYM	Transceiver type For OTG controller core, reset value is '0h'; 000 UTMI/UMTI+ 001 ULPI DDR 010 ULPI 011 Serial
5–4 PHYW	Data width of the transceiver connected to the controller core. For OTG controller core, reset value is '01b'. 00 Reserved 01 16-bit wide data bus [30MHZ clock from the transciever] 10 Reserved 11 Reserved
3–0 -	This field is reserved. Reserved

44.4.3 Host Hardware Parameters (USBx_HWHOST)

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												NPORT			HC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

USBx_HWHOST field descriptions

Field	Description
31–4 -	This field is reserved. Reserved
3–1 NPORT	The Number of downstream ports supported by the host controller is NPORT+1. As all two controller cores are single port, these bits are set to '000b'.
0 HC	Host Capable. Both controller cores support host operation mode. 1 Support host operation mode 0 Not support

44.4.4 Device Hardware Parameters (USBx_HWDEVICE)

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										DEVEP					DC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

USBx_HWDEVICE field descriptions

Field	Description
31–6 -	This field is reserved. Reserved
5–1 DEVEP	Device Endpoint Number OTG controller core has six Endpoints.
0 DC	Device Capable. 1 support device operation mode 0 not support

44.4.5 TX Buffer Hardware Parameters (USBx_HWTXBUF)

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TXCHANADD								Reserved								TXBURST							
W																																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	0

USBx_HWTXBUF field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 TXCHANADD	TX Buffer size is: (2^TXCHANADD) * 4 Bytes. These bits are set to '08h', so buffer size is 256*4 Bytes. For OTG controller core, there is one TX Buffer for each endpoint, so 6 TX buffers in total.
15–8 -	This field is reserved. Reserved
7–0 TXBURST	Default burst size for memory to TX buffer transfer. This is reset value of TXPBURST bits in USB core regsiter UOG_BURSTSIZE. Please see Programmable Burst Size (USB_BURSTSIZE) .

44.4.6 RX Buffer Hardware Parameters (USBx_HWRXBUF)

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																RXADD								RXBURST							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0

USBx_HWRXBUF field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 RXADD	Buffer total size for all receive endpoints is (2^RXADD). RX Buffer size is: (2^RXADD) * 4 Bytes. These bits are set to '08h', so buffer size is 256*4 Bytes. For OTG controller core, there is one RX Buffer, shared by 6 Endpoints.
7–0 RXBURST	Default burst size for memory to RX buffer transfer. This is reset value of RXPBURST bits in USB core regsiter UOG_BURSTSIZE. Please see Programmable Burst Size (USB_BURSTSIZE) .

44.4.7 General Purpose Timer #0 Load (USBx_GPTIMER0LD)

This register controls load value of the count timer in register n_GPTIMER0CTRL. Please see General Purpose Timer #0 Controller (USB_GPTIMER0CTRL) .

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								GPTLD																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_GPTIMER0LD field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–0 GPTLD	General Purpose Timer Load Value These fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. **NOTE:** Max value is 0xFFFFFF or 16.777215 seconds.

44.4.8 General Purpose Timer #0 Controller (USBx_GPTIMER0CTRL)

This register contains the control for this countdown timer and a data field can be queried to determine the running count value. This timer has granularity on 1 us and can be programmed to a little over 16 seconds. There are two counter modes which are described in the register table below. When the timer counter value transitions to zero, an interrupt could be generated if enable.

Interrupt status bit is TI0 bit in n_USBSTS register (See USB Status Register (USB_USBSTS)), interrupt enable bit is TIE0 bit in n_USBINTR register. (See Interrupt Enable Register (USB_USBINTR) .)

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	GPTRUN	GPTRST	Reserved					GPTMODE	GPTCNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	GPTCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_GPTIMER0CTRL field descriptions

Field	Description
31 GPTRUN	General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 Stop counting 1 Run
30 GPTRST	General Purpose Timer Reset 0 No action 1 Load counter value from GPTLD bits in n_GPTIMER0LD
29–25 -	This field is reserved. Reserved
24 GPTMODE	General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software; In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again.

Table continues on the next page...

USBx_GPTIMER0CTRL field descriptions (continued)

Field	Description
	0 One Shot Mode 1 Repeat Mode
23–0 GPTCNT	General Purpose Timer Counter. This field is the count value of the countdown timer.

44.4.9 General Purpose Timer #1 Load (USBx_GPTIMER1LD)

This register controls load value of the count timer in register n_GPTIMER1CTRL. Please see General Purpose Timer #1 Controller (USB_GPTIMER1CTRL) .

Address: Base address + 88h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								GPTLD																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_GPTIMER1LD field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–0 GPTLD	General Purpose Timer Load Value These bit fields are loaded to GPTCNT bits when GPTRST bit is set '1b'. This value represents the time in microseconds minus 1 for the timer duration. Example: for a one millisecond timer, load 1000-1=999 or 0x0003E7. **NOTE:** Max value is 0xFFFFFF or 16.777215 seconds.

44.4.10 General Purpose Timer #1 Controller (USBx_GPTIMER1CTRL)

This register contains the control for this countdown timer and a data field can be queried to determine the running count value. This timer has granularity on 1 us and can be programmed to a little over 16 seconds. There are two counter modes which are described in the register table below. When the timer counter value transitions to zero, an interrupt could be generated if enable.

Interrupt status bit is TI1 bit in UOG_USBSTS register (See USB Status Register (USB_USBSTS)), interrupt enable bit is TIE1 bit in n_USBINTR register (See Interrupt Enable Register (USB_USBINTR)).

Address: Base address + 8Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	GPTRUN	GPTRST	Reserved					GPTMODE	GPTCNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	GPTCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_GPTIMER1CTRL field descriptions

Field	Description
31 GPTRUN	General Purpose Timer Run GPTCNT bits are not effected when setting or clearing this bit. 0 Stop counting 1 Run
30 GPTRST	General Purpose Timer Reset 0 No action 1 Load counter value from GPTLD bits in UOG_GPTIMER0LD
29–25 -	This field is reserved. Reserved
24 GPTMODE	General Purpose Timer Mode In one shot mode, the timer will count down to zero, generate an interrupt, and stop until the counter is reset by software. In repeat mode, the timer will count down to zero, generate an interrupt and automatically reload the counter value from GPTLD bits to start again. 0 One Shot Mode 1 Repeat Mode
23–0 GPTCNT	General Purpose Timer Counter. This field is the count value of the countdown timer.

44.4.11 System Bus Config (USBx_SBUSCFG)

Address: Base address + 90h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																													AHBBRST		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

USBx_SBUSCFG field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
2–0 AHBBRST	AHB master interface Burst configuration These bits controls AHB master transfer type sequence (or priority). **NOTE:** This register overrides n_BURSTSIZE register when its value is not zero. 000 Incremental burst of unspecified length only 001 INCR4 burst, then single transfer 010 INCR8 burst, INCR4 burst, then single transfer 011 INCR16 burst, INCR8 burst, INCR4 burst, then single transfer 100 Reserved, don't use 101 INCR4 burst, then incremental burst of unspecified length 110 INCR8 burst, INCR4 burst, then incremental burst of unspecified length 111 INCR16 burst, INCR8 burst, INCR4 burst, then incremental burst of unspecified length

44.4.12 Capability Register Length (USBx_CAPLENGTH)

The following figure shows Capability Register Length (CAPLENGTH) which indicates the offset that should be added to the register base address at the beginning of the Operational Register.

Address: Base address + 100h offset

Bit	7	6	5	4	3	2	1	0
Read	CAPLENGTH							
Write								
Reset	0	1	0	0	0	0	0	0

USBx_CAPLENGTH field descriptions

Field	Description
7–0 CAPLENGTH	These bits are used as an offset to add to register base to find the beginning of the Operational Register. Default value is '40h'.

44.4.13 Host Controller Interface Version (USBx_HCIVERSION)

The following figure shows the Host Interface version number (HCIVERSION), which is a 2-byte register containing a BCD encoding of the EHCI revision number supported by this host controller. The most significant byte of this register represents a major revision and the least significant byte is the minor revision.

Address: Base address + 102h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	HCIVERSION															
Write																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

USBx_HCIVERSION field descriptions

Field	Description
15–0 HCIVERSION	Host Controller Interface Version Number Default value is '10h', which means EHCI rev1.0.

44.4.14 Host Controller Structural Parameters (USBx_HCSPARAMS)

The following figure shows the port steering logic capabilities of Host Control Structural Parameters (HCSPARAMS).

Address: Base address + 104h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				N_TT				N_PTT				Reserved			PI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	N_CC				N_PCC				Reserved			PPC	N_PORTS			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

USBx_HCSPARAMS field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–24 N_TT	Number of Transaction Translators (N_TT). Default value '0000b' This field indicates the number of embedded transaction translators associated with the USB2.0 host controller. These bits would be set to '0001b' for Multi-Port Host, and '0000b' for Single-Port Host.
23–20 N_PTT	Number of Ports per Transaction Translator (N_PTT). Default value '0000b' This field indicates the number of ports assigned to each transaction translator within the USB2.0 host controller. These bits would be set to equal N_PORTS for Multi-Port Host, and '0000b' for Single-Port Host.
19–17 -	This field is reserved. Reserved
16 PI	Port Indicators (P INDICATOR)
15–12 N_CC	Number of Companion Controller (N_CC). These bits are '0000b' in both the controller cores. 0 There is no internal Companion Controller and port-ownership hand-off is not supported. 1 There are internal companion controller(s) and port-ownership hand-offs is supported.
11–8 N_PCC	Number of Ports per Companion Controller For example, if N_PORTS has a value of 6 and N_CC has a value of 2 then N_PCC could have a value of 3. The convention is that the first N_PCC ports are assumed to be routed to companion controller 1, the next N_PCC ports to companion controller 2, etc. In the previous example, the N_PCC could have been 4, where the first 4 are routed to companion controller 1 and the last two are routed to companion controller 2. The number in this field must be consistent with N_PORTS and N_CC.
7–5 -	This field is reserved. Reserved
4 PPC	Port Power Control This field indicates whether the host controller implementation includes port power control. A one indicates the ports have port power switches. A zero indicates the ports do not have port power switches. The value of this field affects the functionality of the Port Power field in each port status and control register
3–0 N_PORTS	Number of downstream ports. This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register. Valid values are in the range of 1h to Fh. A zero in this field is undefined. These bits are always set to '0001b' because both the controller cores are Single-Port Host.

44.4.15 Host Controller Capability Parameters (USBx_HCCPARAMS)

This register identifies multiple mode control (time-base bit functionality), addressing capability.

Address: Base address + 108h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EECP								IST				0	ASP	PFL	ADC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

USBx_HCCPARAMS field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–8 EECP	EHCI Extended Capabilities Pointer. These bits are set '00h' in both the controller cores.
7–4 IST	Isochronous Scheduling Threshold. These bits are set '00h' in both the controller cores.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 ASP	Asynchronous Schedule Park Capability This bit is set '1b' in both the controller cores.
1 PFL	Programmable Frame List Flag If set to a one, then the system software can specify and use a smaller frame list and configure the host controller via the USBCMD register Frame List Size field. The frame list must always be aligned on a 4K-page boundary. This requirement ensures that the frame list is always physically contiguous.
0 ADC	64-bit Addressing Capability

44.4.16 Device Controller Interface Version (USBx_DCIVERSION)

This register indicates the two-byte BCD encoding of the device controller interface version number.

Address: Base address + 120h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DCIVERSION															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

USBx_DCIVERSION field descriptions

Field	Description
15–0 DCIVERSION	Device Controller Interface Version Number Default value is '01h', which means rev0.1.

44.4.17 Device Controller Capability Parameters (USBx_DCCPARAMS)

These fields describe the overall device capability of the controller.

Address: Base address + 124h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							HC	DC	Reserved		DEN				
W																
Reset	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0

USBx_DCCPARAMS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved

Table continues on the next page...

USBx_DCCPARAMS field descriptions (continued)

Field	Description
8 HC	Host Capable When this bit is 1, this controller is capable of operating as an EHCI compatible USB 2.0 host controller.
7 DC	Device Capable When this bit is 1, this controller is capable of operating as a USB 2.0 device.
6–5 -	This field is reserved. Reserved
4–0 DEN	Device Endpoint Number

44.4.18 USB Command Register (USBx_USBCMD)

The Command Register indicates the command to be executed by the serial bus host/ device controller. Writing to the register causes a command to be executed.

*: ASPE,ASP[1],ASP[0] reset value: '0b' for OTG core.

Address: Base address + 140h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								ITC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	FS2	ATDTW	SUTW	Reserved	ASPE	Reserved	ASP		Reserved	IAA	ASE	PSE	FS1		RST	RS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_USBCMD field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 ITC	Interrupt Threshold Control -Read/Write. The system software uses this field to set the maximum rate at which the host/device controller will issue interrupts. ITC contains the maximum interrupt interval measured in micro-frames. Valid values are shown below. Value Maximum Interrupt Interval 0x00 Immediate (no threshold) 0x01 1 micro-frame 0x02 2 micro-frames 0x04 4 micro-frames 0x08 8 micro-frames 0x10 16 micro-frames 0x20 32 micro-frames 0x40 64 micro-frames
15 FS2	See also bits 3-2 Frame List Size - (Read/Write or Read Only). [host mode only] This field is Read/Write only if Programmable Frame List Flag in the HCCPARAMS registers is set to one. This field specifies the size of the frame list that controls which bits in the Frame Index Register should be used for the Frame List Current index. **NOTE:** This field is made up from USBCMD bits 15, 3 and 2. Value Meaning 000 1024 elements (4096 bytes) Default value 001 512 elements (2048 bytes) 010 256 elements (1024 bytes) 011 128 elements (512 bytes) 100 64 elements (256 bytes) 101 32 elements (128 bytes) 110 16 elements (64 bytes) 111 8 elements (32 bytes)
14 ATDTW	Add dTD TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure proper addition of a new dTD to an active (primed) endpoint's linked list. This bit is set and cleared by software. This bit would also be cleared by hardware when state machine is hazard region for which adding a dTD to a primed endpoint may go unrecognized.
13 SUTW	Setup TripWire - Read/Write. [device mode only] This bit is used as a semaphore to ensure that the setup data payload of 8 bytes is extracted from a QH by the DCD without being corrupted. If the setup lockout mode is off (SLOM bit in USB core register n_USBMODE, see USB Device Mode (USB_USBMODE)) then there is a hazard when new setup data arrives while the DCD is copying the setup data payload from the QH for a previous setup packet. This bit is set and cleared by software. This bit would also be cleared by hardware when a hazard detected.

Table continues on the next page...

USBx_USBCMD field descriptions (continued)

Field	Description
12 -	This field is reserved. Reserved
11 ASPE	Asynchronous Schedule Park Mode Enable - Read/Write. If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this bit defaults to a 1h and is R/W. Otherwise the bit must be a zero and is RO. Software uses this bit to enable or disable Park mode. When this bit is one, Park mode is enabled. When this bit is a zero, Park mode is disabled. This field is set to '1b' in this implementation.
10 -	This field is reserved. Reserved
9–8 ASP	Asynchronous Schedule Park Mode Count - Read/Write. If the *Asynchronous Park Capability* bit in the HCCPARAMS register is a one, then this field defaults to 3h and is R/W. Otherwise it defaults to zero and is Read-Only. It contains a count of the number of successive transactions the host controller is allowed to execute from a high-speed queue head on the Asynchronous schedule before continuing traversal of the Asynchronous schedule. Valid values are 1h to 3h. Software must not write a zero to this bit when *Park Mode Enable* is a one as this will result in undefined behavior. This field is set to 3h in all 4 controller core.
7 -	This field is reserved. Reserved
6 IAA	Interrupt on Async Advance Doorbell - Read/Write. This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule. Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule states, it sets the Interrupt on Async Advance status bit in the USBSTS register. If the Interrupt on Sync Advance Enable bit in the USBINTR register is one, then the host controller will assert an interrupt at the next interrupt threshold. The host controller sets this bit to zero after it has set the Interrupt on Sync Advance status bit in the USBSTS register to one. Software should not write a one to this bit when the asynchronous schedule is inactive. Doing so will yield undefined results. This bit is only used in host mode. Writing a one to this bit when device mode is selected will have undefined results.
5 ASE	Asynchronous Schedule Enable - Read/Write. Default 0b. This bit controls whether the host controller skips processing the Asynchronous Schedule. Only the host controller uses this bit. Values Meaning 0 Do not process the Asynchronous Schedule. 1 Use the ASYNCLISTADDR register to access the Asynchronous Schedule.
4 PSE	Periodic Schedule Enable- Read/Write. Default 0b. This bit controls whether the host controller skips processing the Periodic Schedule. Only the host controller uses this bit. Values Meaning 0 Do not process the Periodic Schedule 1 Use the PERIODICLISTBASE register to access the Periodic Schedule.

Table continues on the next page...

USBx_USBCMD field descriptions (continued)

Field	Description
3–2 FS1	See description at bit 15
1 RST	Controller Reset (RESET) - Read/Write. Software uses this bit to reset the controller. This bit is set to zero by the Host/Device Controller when the reset process is complete. Software cannot terminate the reset process early by writing a zero to this register. Host operation mode: When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports. Software should not set this bit to a one when the HCHalted bit in the USBSTS register is a zero. Attempting to reset an actively running host controller will result in undefined behavior. Device operation mode: When software writes a one to this bit, the Controller resets its internal pipelines, timers, counters, state machines etc. to their initial value. Writing a one to this bit when the device is in the attached state is not recommended, because the effect on an attached host is undefined. In order to ensure that the device is not in an attached state before initiating a device controller reset, all primed endpoints should be flushed and the USBCMD Run/Stop bit should be set to 0.
0 RS	Run/Stop (RS) - Read/Write. Default 0b. 1=Run. 0=Stop. Host operation mode: When set to '1b', the Controller proceeds with the execution of the schedule. The Controller continues execution as long as this bit is set to a one. When this bit is set to 0, the Host Controller completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Controller has finished the transaction and has entered the stopped state. Software should not write a one to this field unless the controller is in the Halted state (that is, HCHalted in the USBSTS register is a one). Device operation mode: Writing a one to this bit will cause the controller to enable a pull-up on D+ and initiate an attach event. This control bit is not directly connected to the pull-up enable, as the pull-up will become disabled upon transitioning into high-speed mode. Software should use this bit to prevent an attach event before the controller has been properly initialized. Writing a 0 to this will cause a detach event.

44.4.19 USB Status Register (USBx_USBSTS)

This register indicates various states of the Host/Device controller and any pending interrupts. This register does not indicate status resulting from a transaction on the serial bus.

Address: Base address + 144h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						TI1	TI0	Reserved							NAKI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	AS	PS	RCL	HCH	Reserved	ULPII	Reserved	SLI	SRI	URI	AAI	SEI	FRI	PCI	UEI	UI
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_USBSTS field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25 TI1	General Purpose Timer Interrupt 1(GPTINT1)--R/WC. This bit is set when the counter in the GPTIMER1CTRL register transitions to zero, writing a one to this bit will clear it.
24 TI0	General Purpose Timer Interrupt 0(GPTINT0)--R/WC. This bit is set when the counter in the GPTIMER0CTRL register transitions to zero, writing a one to this bit clears it.
23–17 -	This field is reserved. Reserved
16 NAKI	NAK Interrupt Bit--RO. This bit is set by hardware when for a particular endpoint both the TX/RX Endpoint NAK bit and corresponding TX/RX Endpoint NAK Enable bit are set. This bit is automatically cleared by hardware when all Enabled TX/RX Endpoint NAK bits are cleared.
15 AS	Asynchronous Schedule Status - Read Only. This bit reports the current real status of the Asynchronous Schedule. When set to zero the asynchronous schedule status is disabled and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Asynchronous Schedule when software transitions the Asynchronous Schedule Enable bit in the USBCMD register. When this bit and the Asynchronous Schedule Enable bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.
14 PS	Periodic Schedule Status - Read Only.

Table continues on the next page...

USBx_USBSTS field descriptions (continued)

Field	Description
	This bit reports the current real status of the Periodic Schedule. When set to zero the periodic schedule is disabled, and if set to one the status is enabled. The Host Controller is not required to immediately disable or enable the Periodic Schedule when software transitions the Periodic Schedule Enable bit in the USBCMD register. When this bit and the Periodic Schedule Enable bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). Only used in the host operation mode.
13 RCL	Reclamation - Read Only. This is a read-only status bit used to detect an empty asynchronous schedule. Only used in the host operation mode.
12 HCH	HCHalted - Read Only. This bit is a zero whenever the Run/Stop bit is a one. The Controller sets this bit to one after it has stopped executing because of the Run/Stop bit being set to 0, either by software or by the Controller hardware (for example, an internal error). Only used in the host operation mode. Default value is '0b' for OTG core. This is because OTG core is not operating as host in default. Please see CM bit in UOG_USBMODE register.
11 -	This field is reserved. Reserved
10 ULPII	ULPI Interrupt - R/WC. This bit will be set '1b' by hardware when there is an event completion in ULPI viewport. This bit is usable only if the controller support UPLI interface mode.
9 -	This field is reserved. Reserved
8 SLI	DCSuspend - R/WC. When a controller enters a suspend state from an active state, this bit will be set to a one. The device controller clears the bit upon exiting from a suspend state. Only used in device operation mode.
7 SRI	SOF Received - R/WC. When the device controller detects a Start Of (micro) Frame, this bit will be set to a one. When a SOF is extremely late, the device controller will automatically set this bit to indicate that an SOF was expected. Therefore, this bit will be set roughly every 1ms in device FS mode and every 125ms in HS mode and will be synchronized to the actual SOF that is received. Because the device controller is initialized to FS before connect, this bit will be set at an interval of 1ms during the prelude to connect and chirp. In host mode, this bit will be set every 125us and can be used by host controller driver as a time base. Software writes a 1 to this bit to clear it.
6 URI	USB Reset Received - R/WC. When the device controller detects a USB Reset and enters the default state, this bit will be set to a one. Software can write a 1 to this bit to clear the USB Reset Received status bit. Only used in device operation mode.
5 AAI	Interrupt on Async Advance - R/WC.

Table continues on the next page...

USBx_USBSTS field descriptions (continued)

Field	Description
	System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the n_USBCMD register. This status bit indicates the assertion of that interrupt source. Only used in host operation mode.
4 SEI	System Error- R/WC. This bit is will be set to '1b' when an Error response is seen to a read on the system interface.
3 FRI	Frame List Rollover - R/WC. The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example. If the frame list size (as programmed in the Frame List Size field of the UOG_USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FHINDEX [12] toggles. Only used in host operation mode.
2 PCI	Port Change Detect - R/WC. The Host Controller sets this bit to a one when on any port a Connect Status occurs, a Port Enable/ Disable Change occurs, or the Force Port Resume bit is set as the result of a J-K transition on the suspended port. The Device Controller sets this bit to a one when the port controller enters the full or high-speed operational state. When the port controller exits the full or high-speed operation states due to Reset or Suspend events, the notification mechanisms are the USB Reset Received bit and the DCSuspend bits respectively.
1 UEI	USB Error Interrupt (USBERRINT) - R/WC. When completion of a USB transaction results in an error condition, this bit is set by the Host/Device Controller. This bit is set along with the USBINT bit, if the TD on which the error interrupt occurred also had its interrupt on complete (IOC) bit set The device controller detects resume signaling only.
0 UI	USB Interrupt (USBINT) - R/WC. This bit is set by the Host/Device Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set. This bit is also set by the Host/Device Controller when a short packet is detected. A short packet is when the actual number of bytes received was less than the expected number of bytes.

44.4.20 Interrupt Enable Register (USBx_USBINTR)

The interrupts to software are enabled with this register. An interrupt is generated when a bit is set and the corresponding interrupt source is active. The USB Status register (n_USBSTS) still shows interrupt sources even if they are disabled by the n_USBINTR register, allowing polling of interrupt events by the software.

Address: Base address + 148h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved						TIE1	TIE0	Reserved				UPIE	UAIE	Reserved	NAKE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	-					ULPIE	Reserved	SLE	SRE	URE	AAE	SEE	FRE	PCE	UEE	UE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_USBINTR field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25 TIE1	General Purpose Timer #1 Interrupt Enable When this bit is one and the TI1 bit in n_USBSTS register is a one the controller will issue an interrupt.
24 TIE0	General Purpose Timer #0 Interrupt Enable When this bit is one and the TI0 bit in n_USBSTS register is a one the controller will issue an interrupt.
23–20 -	This field is reserved. Reserved
19 UPIE	USB Host Periodic Interrupt Enable

Table continues on the next page...

USBx_USBINTR field descriptions (continued)

Field	Description
	When this bit is one, and the UPI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
18 UAIE	USB Host Asynchronous Interrupt Enable When this bit is one, and the UAI bit in the n_USBSTS register is one, host controller will issue an interrupt at the next interrupt threshold.
17 -	This field is reserved. Reserved
16 NAKE	NAK Interrupt Enable When this bit is one and the NAKI bit in n_USBSTS register is a one the controller will issue an interrupt.
15–11 -	These bits are reserved and should be set to zero.
10 ULPIE	ULPI Interrupt Enable This bit is usable only if the controller supports UPLI interface mode.
9 -	This field is reserved. Reserved
8 SLE	Sleep Interrupt Enable When this bit is one and the SLI bit in n_n_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.
7 SRE	SOF Received Interrupt Enable When this bit is one and the SRI bit in n_USBSTS register is a one the controller will issue an interrupt.
6 URE	USB Reset Interrupt Enable When this bit is one and the URI bit in n_UOG_USBSTS register is a one the controller will issue an interrupt. Only used in device operation mode.
5 AAE	Async Advance Interrupt Enable When this bit is one and the AAI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
4 SEE	System Error Interrupt Enable When this bit is one and the SEI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
3 FRE	Frame List Rollover Interrupt Enable When this bit is one and the FRI bit in n_USBSTS register is a one the controller will issue an interrupt. Only used in host operation mode.
2 PCE	Port Change Detect Interrupt Enable When this bit is one and the PCI bit in n_USBSTS register is a one the controller will issue an interrupt.
1 UEE	USB Error Interrupt Enable When this bit is one and the UEI bit in n_USBSTS register is a one the controller will issue an interrupt.
0 UE	USB Interrupt Enalbe When this bit is one and the UI bit in n_USBSTS register is a one the controller will issue an interrupt.

44.4.21 USB Frame Index (USBx_FRINDEX)

This register is used by the host controller to index the periodic frame list. The register updates every 125 microseconds (once each micro-frame). Bits [N: 3] are used to select a particular entry in the Periodic Frame List during periodic schedule execution. The number of bits used for the index depends on the size of the frame list as set by system software in the Frame List Size field in the n_USBCMD register.

This register must be written as a DWord. Byte writes produce-undefined results. This register cannot be written unless the Host Controller is in the 'Halted' state as indicated by the HCHalted bit. A write to this register while the Run/Stop hit is set to a one produces undefined results. Writes to this register also affect the SOF value.

In device mode this register is read only and, the device controller updates the FRINDEX [13:3] register from the frame number indicated by the SOF marker. Whenever a SOF is received by the USB bus, FRINDEX [13:3] will be checked against the SOF marker. If FRINDEX [13:3] is different from the SOF marker, FRINDEX [13:3] will be set to the SOF value and FRINDEX [2:0] will be set to zero (that is, SOF for 1 ms frame). If FRINDEX [13:3] is equal to the SOF value, FRINDEX [2:0] will be increment (that is, SOF for 125 us micro-frame.).

Address: Base address + 14Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																		FRINDEX													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_FRINDEX field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–0 FRINDEX	Frame Index. The value, in this register, increments at the end of each time frame (micro-frame). Bits [N: 3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index. The following illustrates values of N based on the value of the Frame List Size field in the USBCMD register, when used in host mode. USBCMD [Frame List Size] Number Elements N In device mode the value is the current frame number of the last frame transmitted. It is not used as an index. In either mode bits 2:0 indicate the current microframe. 000 (1024) 12 001 (512) 11

Table continues on the next page...

USBx_FRINDEX field descriptions (continued)

Field	Description
	010 (256) 10 011 (128) 9 100 (64) 8 101 (32) 7 110 (16) 6 111 (8) 5

44.4.22 Frame List Base Address (USBx_PERIODICLISTBASE)

Host Controller only

Address: Base address + 154h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	BASEADR																				Reserved											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_PERIODICLISTBASE field descriptions

Field	Description
31–12 BASEADR	Base Address (Low). These bits correspond to memory address signals [31:12], respectively. Only used by the host controller.
11–0 -	This field is reserved. Reserved

44.4.23 Device Address (USBx_DEVICEADDR)

Device Controller only

Address: Base address + 154h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	USBADR							USBADRA	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_DEVICEADDR field descriptions

Field	Description
31–25 USBADR	Device Address. These bits correspond to the USB device address
24 USBADRA	Device Address Advance. Default=0. When this bit is '0', any writes to USBADR are instantaneous. When this bit is written to a '1' at the same time or before USBADR is written, the write to the USBADR field is staged and held in a hidden register. After an IN occurs on endpoint 0 and is ACKed, USBADR will be loaded from the holding register. Hardware will automatically clear this bit on the following conditions: 1) IN is ACKed to endpoint 0. (USBADR is updated from staging register). 2) OUT/SETUP occur to endpoint 0. (USBADR is not updated). 3) Device Reset occurs (USBADR is reset to 0). **NOTE:** After the status phase of the SET_ADDRESS descriptor, the DCD has 2 ms to program the USBADR field. This mechanism will ensure this specification is met when the DCD can not write of the device address within 2ms from the SET_ADDRESS status phase. If the DCD writes the USBADR with USBADRA=1 after the SET_ADDRESS data phase (before the prime of the status phase), the USBADR will be programmed instantly at the correct time and meet the 2ms USB requirement.
23–0 -	This field is reserved. Reserved

44.4.24 Next Asynch. Address (USBx_ASYNCLISTADDR)

Host Controller only

This 32-bit register contains the address of the next asynchronous queue head to be executed by the host. Bits [4:0] of this register cannot be modified by the system software and will always return a zero when read.

Address: Base address + 158h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ASYBASE[31:5]																											Reserved				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_ASYNCLISTADDR field descriptions

Field	Description
31–5 ASYBASE[31:5]	Link Pointer Low (LPL). These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH). Only used by the host controller.
4–0 -	This field is reserved. Reserved

44.4.25 Endpoint List Address (USBx_ENDPTLISTADDR)

Device Controller only

In device mode, this register contains the address of the top of the endpoint list in system memory. Bits [10:0] of this register cannot be modified by the system software and will always return a zero when read.

The memory structure referenced by this physical memory pointer is assumed 64-byte.

Address: Base address + 158h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EPBASE[31:11]																					Reserved										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_ENDPTLISTADDR field descriptions

Field	Description
31–11 EPBASE[31:11]	Endpoint List Pointer(Low). These bits correspond to memory address signals [31:11], respectively. This field will reference a list of up to 32 Queue Head (OH) (that is, one queue head per endpoint & direction).
10–0 -	This field is reserved. Reserved

44.4.26 Programmable Burst Size (USBx_BURSTSIZE)

This register is used to control the burst size used during data movement on the AHB master interface. This register is ignored if AHBBRST bits in SBUSCFG register is non-zero value.

Address: Base address + 160h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															TXPBURST									RXPBURST							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

USBx_BURSTSIZE field descriptions

Field	Description
31–17 -	This field is reserved. Reserved
16–8 TXPBURST	Programmable TX Burst Size. Default value is determined by TXBURST bits in n_HWTXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from system memory to the USB bus.
7–0 RXPBURST	Programmable RX Burst Size. Default value is determined by TXBURST bits in n_HWRXBUF. This register represents the maximum length of a the burst in 32-bit words while moving data from the USB bus to system memory.

44.4.27 TX FIFO Fill Tuning (USBx_TXFILLTUNING)

The fields in this register control performance tuning associated with how the host controller posts data to the TX latency FIFO before moving the data onto the USB bus. The specific areas of performance include the how much data to post into the FIFO and an estimate for how long that operation should take in the target system.

Definitions:

T_0 = Standard packet overhead

T_1 = Time to send data payload

T_{ff} = Time to fetch packet into TX FIFO up to specified level.

T_s = Total Packet Flight Time (send-only) packet

$T_s = T_0 + T_1$

T_p = Total Packet Time (fetch and send) packet

$T_p = T_{ff} + T_0 + T_1$

Upon discovery of a transmit (OUT/SETUP) packet in the data structures, host controller checks to ensure T_p remains before the end of the [micro]frame. If so it proceeds to pre-fill the TX FIFO. If at anytime during the pre-fill operation the time remaining the [micro]frame is < T_s then the packet attempt ceases and the packet is tried at a later time. Although this is not an error condition and the host controller will eventually recover, a mark will be made the scheduler health counter to note the occurrence of a "back-off" event. When a back-off event is detected, the partial packet fetched may need to be discarded from the latency buffer to make room for periodic traffic that will begin after the next SOF. Too many back-off events can waste bandwidth and power on the system bus and thus should be minimized (not necessarily eliminated). Back-offs can be minimized with use of the n_TSCHHEALTH (T_{ff}) described below.

Address: Base address + 164h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TXFIFOTHRES						Reserved			TXSCHHEALTH					TXSCHOH							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

USBx_TXFILLTUNING field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–16 TXFIFOTHRES	FIFO Burst Threshold. (Read/Write) This register controls the number of data bursts that are posted to the TX latency FIFO in host mode before the packet begins on to the bus. The minimum value is 2 and this value should be a low as possible to maximize USB performance. A higher value can be used in systems with unpredictable latency and/or insufficient bandwidth where the FIFO may underrun because the data transferred from the latency FIFO to USB occurs before it can be replenished from system memory. This value is ignored if the Stream Disable bit in UOG_USBMODE register is set. Default value is '00h' for OTG controller core.
15–13 -	This field is reserved. Reserved
12–8 TXSCHHEALTH	Scheduler Health Counter. (Read/Write To Clear) This register increments when the host controller fails to fill the TX latency FIFO to the level programmed by TXFIFOTHRES before running out of time to send the packet before the next Start-Of-Frame. This health counter measures the number of times this occurs to provide feedback to selecting a proper TXSCHOH. Writing to this register will clear the counter and this counter will max. at 31. Default value is '08h' for OTG controller core.
7–0 TXSCHOH	Scheduler Overhead. (Read/Write) [Default = 0] This register adds an additional fixed offset to the schedule time estimator described above as Tff. As an approximation, the value chosen for this register should limit the number of back-off events captured in the TXSCHHEALTH to less than 10 per second in a highly utilized bus. Choosing a value that is too high for

Table continues on the next page...

USBx_TXFILLTUNING field descriptions (continued)

Field	Description
	this register is not desired as it can needlessly reduce USB utilization. The time unit represented in this register is 1.267us when a device is connected in High-Speed Mode. The time unit represented in this register is 6.333us when a device is connected in Low/Full Speed Mode. Default value is '08h' for OTG controller core.

44.4.28 Endpoint NAK (USBx_ENDPTNAK)

Address: Base address + 178h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										EPTN[5:0]						Reserved										EPRN[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_ENDPTNAK field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–16 EPTN[5:0]	TX Endpoint NAK - R/WC. Each TX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received IN token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-5
15–6 -	This field is reserved. Reserved
5–0 EPRN[5:0]	RX Endpoint NAK - R/WC. Each RX endpoint has 1 bit in this field. The bit is set when the device sends a NAK handshake on a received OUT or PING token for the corresponding endpoint. Bit [N] - Endpoint #[N], N is 0-5

44.4.29 Endpoint NAK Enable (USBx_ENDPTNAKEN)

Address: Base address + 17Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										EPTNE[5:0]						Reserved										EPRNE[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_ENDPTNAKEN field descriptions

Field	Description
31–22 -	This field is reserved. Reserved

Table continues on the next page...

USBx_ENDPTNAKEN field descriptions (continued)

Field	Description
21–16 EPTNE[5:0]	TX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding TX Endpoint NAK bit. If this bit is set and the corresponding TX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit [N] - Endpoint #[N], N is 0-5
15–6 -	This field is reserved. Reserved
5–0 EPRNE[5:0]	RX Endpoint NAK Enable - R/W. Each bit is an enable bit for the corresponding RX Endpoint NAK bit. If this bit is set and the corresponding RX Endpoint NAK bit is set, the NAK Interrupt bit is set. Bit [N] - Endpoint #[N], N is 0-5

44.4.30 Port Status & Control (USBx_PORTSC1)

Host Controller

A host controller could implement one to eight port status and control registers. The number is determined by N_PORTs bits in HWSPARAMs register (please see Host Controller Structural Parameters (USB_HCSPARAMS)). Software could read this parameter register to determine how many ports need service.

Both the controller cores are Single-Port Host, so there is only one port status and control register for each controller core.

This register is only reset by power on reset or controller core reset. The initial conditions of a port are:

- No device connected
- Port disabled

If the port supports power control, this state remains until port power is supplied (by software).

Device Controller

A device controller has only port register one (PORTSC1) and it does not support power control. Port control in device mode is only used for status port reset, suspend, resume, and current connect status. It is also used to initiate test mode or force signaling and allows software to put the PHY into low power suspend mode and disable the PHY clock.

Address: Base address + 184h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	PTS2		STS	PTW	PSPD		PTS1	PFSC	PHCD	WKOC	WKDC	WKCN	PTC[3:0]			
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	PIC[1:0]		PO	PP	LS[1:0]		HSP	PR	SUSP	FPR	OCC	OCA	PEC	PE	CSC	CCS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_PORTSC1 field descriptions

Field	Description
31–30 PTS2	Parallel Transceiver Select (bit25, bit31, bit30). These register bits are implementation dependent. Field {bit25, bit31, bit30}. This need to be set to 3'b000. As such, its reset value is 3'b000.
29 STS	1 Serial Interface Engine is selected When this bit is set '1b', serial interface engine will be used instead of parallel interface signals.
28 PTW	Parallel Transceiver Width This bit has no effect if serial interface engine is used. These register bits are implementation dependent. It is Read-Only. Reset value is '1b'. 0 Reserved 1 Select the 16-bit UTMI interface [30 MHz]
27–26 PSPD	Port Speed - Read Only. This register field indicates the speed at which the port is operating. 00 Full Speed 01 Low Speed 10 High Speed 11 Undefined
25 PTS1	See description at bits 31-30
24 PFSC	Port Force Full Speed Connect - Read/Write. Default = 0b. When this bit is set to '1b', the port will be forced to only connect at Full Speed, It disables the chirp sequence that allows the port to identify itself as High Speed. 1 Forced to full speed 0 Normal operation
23 PHCD	PHY Low Power Suspend - Clock Disable (PLPSCD) - Read/Write. Default = 0b. When this bit is set to '1b', the PHY clock is disabled. Reading this bit will indicate the status of the PHY clock. **NOTE:** The PHY clock cannot be disabled if it is being used as the system clock.

Table continues on the next page...

USBx_PORTSC1 field descriptions (continued)

Field	Description
	In device mode, The PHY can be put into Low Power Suspend when the device is not running (USBCMD Run/Stop=0b) or the host has signaled suspend (PORTSC1 SUSPEND=1b). PHY Low power suspend will be cleared automatically when the host initials resume. Before forcing a resume from the device, the device controller driver must clear this bit. In host mode, the PHY can be put into Low Power Suspend when the downstream device has been put into suspend mode or when no downstream device is connected. Low power suspend is completely under the control of software. 1 Disable PHY clock 0 Enable PHY clock
22 WKOC	Wake on Over-current Enable (WKOC_E) - Read/Write. Default = 0b. Writing this bit to a one enables the port to be sensitive to over-current conditions as wake-up events. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero.
21 WKDC	Wake on Disconnect Enable (WKDSCNNT_E) - Read/Write. Default=0b. Writing this bit to a one enables the port to be sensitive to device disconnects as wake-up events. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero or in device mode.
20 WKCN	Wake on Connect Enable (WKCNNT_E) - Read/Write. Default=0b. Writing this bit to a one enables the port to be sensitive to device connects as wake-up events. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero or in device mode.
19–16 PTC[3:0]	Port Test Control - Read/Write. Default = 0000b. Refer to Section 4.14 for the operational model for using these test modes and the USB Specification Revision 2.0, Chapter 7 for details on each test mode. The FORCE_ENABLE_FS and FORCE ENABLE_LS are extensions to the test mode support specified in the EHCI specification. Writing the PTC field to any of the FORCE_ENABLE_{HS/FS/LS} values will force the port into the connected and enabled state at the selected speed. Writing the PTC field back to TEST_MODE_DISABLE will allow the port state machines to progress normally from that point. **NOTE:** *Low speed operations are not supported as a peripheral device.* Any other value than zero indicates that the port is operating in test mode. Value Specific Test 0000 TEST_MODE_DISABLE 0001 J_STATE 0010 K_STATE 0011 SE0 (host) / NAK (device) 0100 Packet 0101 FORCE_ENABLE_HS 0110 FORCE_ENABLE_FS 0111 FORCE_ENABLE_LS 1000-1111 Reserved
15–14 PIC[1:0]	Port Indicator Control - Read/Write. Default = Ob. Writing to this field has no effect if the P_INDICATOR bit in the HCSPARAMS register is a zero. Refer to the USB Specification Revision 2.0 for a description on how these bits are to be used. This field is zero if *Port Power* is zero. Bit Value Meaning

Table continues on the next page...

USBx_PORTSC1 field descriptions (continued)

Field	Description
	00 Port indicators are off 01 Amber 10 Green 11 Undefined
13 PO	Port Owner-Read/Write. Always = 0. This bit unconditionally goes to a 0 when the configured bit in the CONFIGFLAG register makes a 0 to 1 transition. This bit unconditionally goes to 1 whenever the Configured bit is zero System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a one to this bit when the attached device is not a high-speed device. A one in this bit means that an internal companion controller owns and controls the port. Port owner handoff is not supported, therefore this bit will always be 0.
12 PP	Port Power (PP)-Read/Write or Read Only. The function of this bit depends on the value of the Port Power Switching (PPC) field in the HCSPARAMS register. The behavior is as follows: PPC PP Operation 0 1b *Read Only - Host controller does not have port power control switches. Each port is hard-wired to power.* 1 1b/0b - *Read/Write. Host/OTG controller requires port power control switches. This bit represents the current setting of the switch (0=off, 1=on). When power is not available on a port (that is, PP equals a 0), the port is non-functional and will not report attaches, detaches, etc.* When an over-current condition is detected on a powered port and PPC is a one, the PP bit in each affected port may be transitional by the host controller driver from a one to a zero (removing power from the port). This feature is implemented in both the controller cores (PPC = 1).
11–10 LS[1:0]	Line Status-Read Only. These bits reflect the current logical levels of the D+ (bit 11) and D- (bit 10) signal lines. In host mode, the use of linestate by the host controller driver is not necessary (unlike EHCI), because the port controller state machine and the port routing manage the connection of LS and FS. In device mode, the use of linestate by the device controller driver is not necessary. The encoding of the bits are: Bits [11:10] Meaning 00 SE0 10 J-state 01 K-state 11 Undefined
9 HSP	High-Speed Port - Read Only. Default = 0b. When the bit is one, the host/device connected to the port is in high-speed mode and if set to zero, the host/device connected to the port is not in a high-speed mode. **NOTE:** HSP is redundant with PSPD(bit 27, 26) but remained for compatibility.
8 PR	Port Reset - Read/Write or Read Only. Default = 0b. In Host Mode: Read/Write. 1=Port is in Reset. 0=Port is not in Reset. Default 0.

Table continues on the next page...

USBx_PORTSC1 field descriptions (continued)

Field	Description
	When software writes a one to this bit the bus-reset sequence as defined in the USB Specification Revision 2.0 is started. *This bit will automatically change to zero after the reset sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the reset duration is timed in the driver.* In Device Mode: This bit is a read only status bit. Device reset from the USB bus is also indicated in the USBSTS register. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero.
7 SUSP	Suspend - Read/Write or Read Only. Default = 0b. 1=Port in suspend state. 0=Port not in suspend state. In Host Mode: Read/Write. Port Enabled Bit and Suspend bit of this register define the port states as follows: Bits [Port Enabled, Suspend] Port State 0x Disable 10 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for port reset. The blocking occurs at the end of the current transaction if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress. The host controller will unconditionally set this bit to zero when software sets the *Force Port Resume* bit to zero. The host controller ignores a write of zero to this bit. If host software sets this bit to a one when the port is not enabled (that is, *Port enabled* bit is a zero) the results are undefined. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero in host mode. In Device Mode: Read Only. In device mode this bit is a read only status bit.
6 FPR	Force Port Resume -Read/Write. 1= Resume detected/driven on port. 0=No resume (K-state) detected/ driven on port. Default = 0. In Host Mode: Software sets this bit to one to drive resume signaling. The Host Controller sets this bit to one if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a one because a J-to-K transition is detected, the *Port Change Detect* bit in the USBSTS register is also set to one. *This bit will automatically change to zero after the resume sequence is complete. This behavior is different from EHCI where the host controller driver is required to set this bit to a zero after the resume duration is timed in the driver.* Note that when the Host controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a one. This bit will remain a one until the port has switched to the high-speed idle. Writing a zero has no affect because the port controller will time the resume operation and clear the bit when the port control state switches to HS or FS idle. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero in host mode. This bit is not-EHCI compatible. In Device mode:

Table continues on the next page...

USBx_PORTSC1 field descriptions (continued)

Field	Description
	After the device has been in Suspend State for 5ms or more, software must set this bit to one to drive resume signaling before clearing. The Device Controller will set this bit to one if a J-to-K transition is detected while the port is in the Suspend state. The bit will be cleared when the device returns to normal operation. Also, when this bit transitions to a one because a J-to-K transition detected, the *Port Change Detect* bit in the USBSTS register is also set to one.
5 OCC	Over-current Change-R/WC. Default=0. This bit is set '1b' by hardware when there is a change to Over-current Active. Software can clear this bit by writing a one to this bit position.
4 OCA	Over-current Active-Read Only. Default 0. This bit will automatically transition from one to zero when the over current condition is removed. 1 This port currently has an over-current condition 0 This port does not have an over-current condition.
3 PEC	Port Enable/Disable Change-R/WC. 1=Port enabled/disabled status has changed. 0=No change. Default = 0. In Host Mode: For the root hub, this bit is set to a one only when a port is disabled due to disconnect on the port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification). Software clears this by writing a one to it. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero. In Device mode: The device port is always enabled, so this bit is always '0b'.
2 PE	Port Enabled/Disabled-Read/Write. 1=Enable. 0=Disable. Default 0. In Host Mode: Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a one to this field. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by the host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events. When the port is disabled, (0b) downstream propagation of data is blocked except for reset. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero in host mode. In Device Mode: The device port is always enabled, so this bit is always '1b'.
1 CSC	Connect Status Change-R/WC. 1 =Change in Current Connect Status. 0=No change. Default 0. In Host Mode: Indicates a change has occurred in the port's Current Connect Status. The host/device controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be 'setting' an already-set bit (that is, the bit will remain set). Software clears this bit by writing a one to it. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero in host mode. In Device Mode: This bit is undefined in device controller mode.

Table continues on the next page...

USBx_PORTSC1 field descriptions (continued)

Field	Description
0 CCS	Current Connect Status-Read Only. In Host Mode: 1=Device is present on port. 0=No device is present. Default = 0. This value reflects the current state of the port, and may not correspond directly to the event that caused the *Connect Status Change* bit (Bit 1) to be set. This field is zero if *Port Power(*Port Status & Control (USB_PORTSC1)*)* is zero in host mode. In Device Mode: 1=Attached. 0=Not Attached. Default=0. A one indicates that the device successfully attached and is operating in either high speed or full speed as indicated by the High Speed Port bit in this register. A zero indicates that the device did not attach successfully or was forcibly disconnected by the software writing a zero to the Run bit in the USBCMD register. It does not state the device being disconnected or suspended.

44.4.31 On-The-Go Status & control (USBx_OTGSC)

It has four sections:

- OTG Interrupt enables (Read/Write)
- OTG Interrupt status (Read/Write to Clear)
- OTG Status inputs (Read Only)
- OTG Controls (Read/Write)

The status inputs are debounced using a 1 ms time constant. Values on the status inputs that do not persist for more than 1 ms do not cause an update of the status input register, or cause an OTG interrupt.

See also USB Device Mode (USB_USBMODE) register.

Address: Base address + 1A4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	DPIE	1MSSE	BSEIE	BSVIE	ASVIE	AVVIE	IDIE	Reserved	DPIS	1MSS	BSEIS	BSVIS	ASVIS	AVVIS	IDIS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	DPS	1msT	BSE	BSV	ASV	AVV	ID	Reserved		IDPU	0	OT	Reserved	0	VD
W																
Reset	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0

USBx_OTGSC field descriptions

Field	Description
31 -	This field is reserved. Reserved
30 DPIE	Data Pulse Interrupt Enable
29 1MSSE	1 millisecond timer Interrupt Enable - Read/Write
28 BSEIE	B Session End Interrupt Enable - Read/Write. Setting this bit enables the B session end interrupt.
27 BSVIE	B Session Valid Interrupt Enable - Read/Write. Setting this bit enables the B session valid interrupt.
26 ASVIE	A Session Valid Interrupt Enable - Read/Write. Setting this bit enables the A session valid interrupt.
25 AVVIE	A VBus Valid Interrupt Enable - Read/Write. Setting this bit enables the A VBus valid interrupt.
24 IDIE	USB ID Interrupt Enable - Read/Write. Setting this bit enables the USB ID interrupt.
23 -	This field is reserved. Reserved
22 DPIS	Data Pulse Interrupt Status - Read/Write to Clear. This bit is set when data bus pulsing occurs on DP or DM. Data bus pulsing is only detected when USBMODE.CM = Host (11) and PORTSC1(0)[PP] = 0. Software must write a one to clear this bit.
21 1MSS	1 millisecond timer Interrupt Status - Read/Write to Clear. This bit is set once every millisecond. Software must write a one to clear this bit.
20 BSEIS	B Session End Interrupt Status - Read/Write to Clear. This bit is set when VBus has fallen below the B session end threshold. Software must write a one to clear this bit
19 BSVIS	B Session Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the B session valid threshold (0.8 VDC).

Table continues on the next page...

USBx_OTGSC field descriptions (continued)

Field	Description
	Software must write a one to clear this bit.
18 ASVIS	A Session Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the A session valid threshold (0.8 VDC). Software must write a one to clear this bit.
17 AVVIS	A VBus Valid Interrupt Status - Read/Write to Clear. This bit is set when VBus has either risen above or fallen below the VBus valid threshold (4.4 VDC) on an A device. Software must write a one to clear this bit.
16 IDIS	USB ID Interrupt Status - Read/Write. This bit is set when a change on the ID input has been detected. Software must write a one to clear this bit.
15 -	This field is reserved. Reserved
14 DPS	Data Bus Pulsing Status - Read Only. A '1' indicates data bus pulsing is being detected on the port.
13 1msT	1 millisecond timer toggle - Read Only. This bit toggles once per millisecond.
12 BSE	B Session End - Read Only. Indicates VBus is below the B session end threshold.
11 BSV	B Session Valid - Read Only. Indicates VBus is above the B session valid threshold.
10 ASV	A Session Valid - Read Only. Indicates VBus is above the A session valid threshold.
9 AVV	A VBus Valid - Read Only. Indicates VBus is above the A VBus valid threshold.
8 ID	USB ID - Read Only. 0 = A device, 1 = B device
7–6 -	This field is reserved. Reserved
5 IDPU	ID Pullup - Read/Write This bit provide control over the ID pull-up resister; 0 = off, 1 = on [default]. When this bit is 0, the ID input will not be sampled.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 OT	OTG Termination - Read/Write. This bit must be set when the OTG device is in device mode, this controls the pulldown on DM.
2 -	This field is reserved. Reserved
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

USBx_OTGSC field descriptions (continued)

Field	Description
0 VD	VBUS_Discharge - Read/Write. Setting this bit causes VBus to discharge through a resistor.

44.4.32 USB Device Mode (USBx_USBMODE)

Address: Base address + 1A8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved										SDIS	SLOM	ES	CM	
W																
Reset	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0

USBx_USBMODE field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14–5 -	This field is reserved. Reserved
4 SDIS	Stream Disable Mode. (0 - Inactive [default]; 1 - Active) Device Mode: Setting to a '1' disables double priming on both RX and TX for low bandwidth systems. This mode ensures that when the RX and TX buffers are sufficient to contain an entire packet that the standard double buffering scheme is disabled to prevent overruns/underruns in bandwidth limited systems. Note: In High Speed Mode, all packets received are responded to with a NYET handshake when stream disable is active.

Table continues on the next page...

USBx_USBMODE field descriptions (continued)

Field	Description
	Host Mode: Setting to a '1' ensures that overruns/underruns of the latency FIFO are eliminated for low bandwidth systems where the RX and TX buffers are sufficient to contain the entire packet. Enabling stream disable also has the effect of ensuring the TX latency is filled to capacity before the packet is launched onto the USB. **NOTE:** Time duration to pre-fill the FIFO becomes significant when stream disable is active. See TX FIFO Fill Tuning (USB_TXFILLTUNING) and TXTTFILLTUNING [MPH Only] to characterize the adjustments needed for the scheduler when using this feature. **NOTE:** The use of this feature substantially limits of the overall USB performance that can be achieved.
3 SLOM	Setup Lockout Mode. In device mode, this bit controls behavior of the setup lock mechanism. See Control Endpoint Operation Model . 0 Setup Lockouts On (default); 1 Setup Lockouts Off (DCD requires use of Setup Data Buffer Tripwire in USB Command Register (USB_USBCMD) .
2 ES	Endian Select - Read/Write. This bit can change the byte alignment of the transfer buffers to match the host microprocessor. The field in the microprocessor interface and the data structures are unaffected by the value of this bit because they are based upon the 32-bit word. Bit Meaning 0 Little Endian [Default] 1 Big Endian
1–0 CM	Controller Mode - R/WO. Controller mode is defaulted to the proper mode for host only and device only implementations. For those designs that contain both host & device capability, the controller defaults to an idle state and needs to be initialized to the desired operating mode after reset. For combination host/ device controllers, this register can only be written once after reset. If it is necessary to switch modes, software must reset the controller by writing to the *RESET* bit in the USBCMD register before reprogramming this register. For OTG controller core, reset value is '10b'. Bit Meaning 00 Idle [Default for combination host/device] 01 Reserved 10 Device Controller [Default for device only controller] 11 Host Controller [Default for host only controller]

44.4.33 Endpoint Setup Status (USBx_ENDPTSETUPSTAT)

Address: Base address + 1ACh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ENDPTSETUPSTAT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_ENDPTSETUPSTAT field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15–0 ENDPTSETUPSTAT	Setup Endpoint Status. For every setup transaction that is received, a corresponding bit in this register is set to one. Software must clear or acknowledge the setup transfer by writing a one to a respective bit after it has read the setup data from Queue head. The response to a setup packet as in the order of operations and total response time is crucial to limit bus time outs while the setup lock our mechanism is engaged. See Managing Endpoints in the Device Operational Model. This register is only used in device mode.

44.4.34 Endpoint Initialization (USBx_ENDPTPRIME)

This register is used only in device mode.

Address: Base address + 1B0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										PETB[5:0]						Reserved										PERB[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_ENDPTPRIME field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–16 PETB[5:0]	Prime Endpoint Transmit Buffer - R/WS. For each endpoint a corresponding bit is used to request that a buffer prepared for a transmit operation in order to respond to a USB IN/INTERRUPT transaction. Software should write a one to the corresponding bit when posting a new transfer descriptor to an endpoint. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a transmit buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. **NOTE:** These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PETB[N] - Endpoint #N, N is in 0..5
15–6 -	This field is reserved. Reserved
5–0 PERB[5:0]	Prime Endpoint Receive Buffer - R/WS. For each endpoint, a corresponding bit is used to request a buffer prepare for a receive operation for when a USB host initiates a USB OUT transaction. Software should write a one to the corresponding bit whenever posting a new transfer descriptor to an endpoint. Hardware automatically uses this bit to begin parsing for a new transfer descriptor from the queue head and prepare a receive buffer. Hardware clears this bit when the associated endpoint(s) is (are) successfully primed. **NOTE:** These bits are momentarily set by hardware during hardware re-priming operations when a dTD is retired, and the dQH is updated. PERB[N] - Endpoint #N, N is in 0..5

44.4.35 Endpoint De-Initialize (USBx_ENDPTFLUSH)

This register is used only in device mode.

Address: Base address + 1B4h offset

Bit	31–22	21–16	15–6	5–0
R/W	Reserved	FETB[5:0]	Reserved	FERB[5:0]
Reset	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0

USBx_ENDPTFLUSH field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–16 FETB[5:0]	Flush Endpoint Transmit Buffer - R/WS. Writing one to a bit(s) in this register causes the associated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FETB[N] - Endpoint #N, N is in 0..5
15–6 -	This field is reserved. Reserved
5–0 FERB[5:0]	Flush Endpoint Receive Buffer - R/WS. Writing one to a bit(s) causes the assocUOGiated endpoint(s) to clear any primed buffers. If a packet is in progress for one of the associated endpoints, then that transfer continues until completion. Hardware clears this register after the endpoint flush operation is successful. FERB[N] - Endpoint #N, N is in 0..5

44.4.36 Endpoint Status (USBx_ENDPTSTAT)

This register is used only in device mode.

Address: Base address + 1B8h offset

Bit	31–22	21–16	15–6	5–0
R	Reserved	ETBR[5:0]	Reserved	ERBR[5:0]
W				
Reset	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0

USBx_ENDPTSTAT field descriptions

Field	Description
31–22 -	This field is reserved. Reserved

Table continues on the next page...

USBx_ENDPTSTAT field descriptions (continued)

Field	Description
21–16 ETBR[5:0]	Endpoint Transmit Buffer Ready -- Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to one by the hardware as a response to receiving a command from a corresponding bit in the ENDPTPRIME register. There is always a delay between setting a bit in the ENDPTPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. **NOTE:** These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ETBR[N] - Endpoint #N, N is in 0..5
15–6 -	This field is reserved. Reserved
5–0 ERBR[5:0]	Endpoint Receive Buffer Ready -- Read Only. One bit for each endpoint indicates status of the respective endpoint buffer. This bit is set to a one by the hardware as a response to receiving a command from a corresponding bit in the ENDPRIME register. There is always a delay between setting a bit in the ENDPRIME register and endpoint indicating ready. This delay time varies based upon the current USB traffic and the number of bits set in the ENDPRIME register. Buffer ready is cleared by USB reset, by the USB DMA system, or through the ENDPTFLUSH register. **NOTE:** These bits are momentarily cleared by hardware during hardware endpoint re-priming operations when a dTD is retired, and the dQH is updated. ERBR[N] - Endpoint #N, N is in 0..5

44.4.37 Endpoint Complete (USBx_ENDPTCOMPLETE)

This register is used only in device mode.

Address: Base address + 1BCh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										ETCE						Reserved										ERCE					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_ENDPTCOMPLETE field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–16 ETCE	Endpoint Transmit Complete Event - R/WC. Each bit indicates a transmit event (IN/INTERRUPT) occurred and software should read the corresponding endpoint queue to determine the endpoint status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the *USBINT* . Writing one clears the corresponding bit in this register. ETCE[N] - Endpoint #N, N is in 0..5
15–6 -	This field is reserved. Reserved

Table continues on the next page...

USBx_ENDPTCOMPLETE field descriptions (continued)

Field	Description
5–0 ERCE	Endpoint Receive Complete Event - RW/C. Each bit indicates a received event (OUT/SETUP) occurred and software should read the corresponding endpoint queue to determine the transfer status. If the corresponding IOC bit is set in the Transfer Descriptor, then this bit is set simultaneously with the *USBINT* . Writing one clears the corresponding bit in this register. ERCE[N] - Endpoint #N, N is in 0..5

44.4.38 Endpoint Control0 (USBx_ENDPTCTRL0)

Every device implements Endpoint0 as a control endpoint.

Address: Base address + 1C0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								TXE	Reserved			TXT		Reserved	TXS
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved								RXE	Reserved			RXT		Reserved	RXS
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

USBx_ENDPTCTRL0 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 1 Enabled Endpoint0 is always enabled.

Table continues on the next page...

USBx_ENDPTCTRL0 field descriptions (continued)

Field	Description
22–20 -	This field is reserved. Reserved
19–18 TXT	TX Endpoint Type - Read/Write 00 - Control Endpoint0 is fixed as a Control End Point.
17 -	This field is reserved. Reserved
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK [Default] 1 End Point Stalled Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues returning STALL until the bit is cleared by software or it is automatically cleared upon receipt of a new SETUP request.
15–8 -	This field is reserved. Reserved
7 RXE	RX Endpoint Enable 1 Enabled Endpoint0 is always enabled.
6–4 -	This field is reserved. Reserved
3–2 RXT	RX Endpoint Type - Read/Write 00 Control Endpoint0 is fixed as a Control End Point.
1 -	This field is reserved. Reserved
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues returning STALL until the bit is cleared by software or it is automatically cleared upon receipt of a new SETUP request.

44.4.39 Endpoint Controln (USB*x*_ENDPTCTRL*n*)

There is a UOG_ENDPTCTRLx register for each endpoint in a device.

NOTE

If one endpoint direction is enabled and the paired endpoint of opposite direction is disabled then the unused direction type must be changed from the default control-type to any other type (that is bulk, isochronous, or interrupt-types). Leaving an

unconfigured endpoint control causes undefined behavior for the data pid tracking on the active endpoint/direction.

Address: Base address + 1C4h offset + (4d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								TXE	TXR	TXI	Reserved	TXT		TXD	TXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved								RXE	RXR	RXI	Reserved	RXT		RXD	RXS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBx_ENDPTCTRL*n* field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 TXE	TX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
22 TXR	TX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the Host and device.
21 TXI	TX Data Toggle Inhibit 0 PID Sequencing Enabled. [Default] 1 PID Sequencing Disabled. This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always transmit DATA0 for a data packet.
20 -	This field is reserved. Reserved

Table continues on the next page...

USBx_ENDPTCTRL*n* field descriptions (continued)

Field	Description
19–18 TXT	TX Endpoint Type - Read/Write 00 Control 01 Isochronous 10 Bulk 11 Interrupt
17 TXD	TX Endpoint Data Source - Read/Write 0 Dual Port Memory Buffer/DMA Engine [DEFAULT] Should always be written as 0.
16 TXS	TX Endpoint Stall - Read/Write 0 End Point OK 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is not configured as a Control Endpoint. It is cleared automatically upon receipt of a SETUP request if this Endpoint is configured as a Control Endpoint. Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues to returning STALL until this bit is either cleared by software or automatically cleared as above. **NOTE:** For CONTROL type endpoint, there is a slight delay (50 clocks max) between the ENDPTSETUPSTAT begin cleared and hardware continuing to clear this bit. In most systems, it is unlikely the DCD software will observe this delay. Take care that the STALL bit is not set immediately after writing a '1' to it. Please follow this procedure: continually write this STALL bit until it is set or until a new setup has ben received by checking the associated ENDPTSETUPSTAT bit.
15–8 -	This field is reserved. Reserved
7 RXE	RX Endpoint Enable 0 Disabled [Default] 1 Enabled An Endpoint should be enabled only after it has been configured.
6 RXR	RX Data Toggle Reset (WS) Write 1 - Reset PID Sequence Whenever a configuration event is received for this Endpoint, software must write a one to this bit in order to synchronize the data PID's between the host and device.
5 RXI	RX Data Toggle Inhibit 0 Disabled [Default] 1 Enabled This bit is only used for test and should always be written as zero. Writing a one to this bit causes this endpoint to ignore the data toggle sequence and always accept data packet regardless of their data PID.
4 -	This field is reserved. Reserved.
3–2 RXT	RX Endpoint Type - Read/Write 00 Control

Table continues on the next page...

USBx_ENDPTCTRL*n* field descriptions (continued)

Field	Description
	01 Isochronous 10 Bulk 11 Reserved
1 RXD	RX Endpoint Data Sink - Read/Write - TBD 0 Dual Port Memory Buffer/DMA Engine [Default] Should always be written as zero.
0 RXS	RX Endpoint Stall - Read/Write 0 End Point OK. [Default] 1 End Point Stalled This bit is set automatically upon receipt of a SETUP request if this Endpoint is not configured as a Control Endpoint. It is cleared automatically upon receipt a SETUP request if this Endpoint is configured as a Control Endpoint, Software can write a one to this bit to force the endpoint to return a STALL handshake to the Host. It continues to returning STALL until this bit is either cleared by software or automatically cleared as above.

44.5 Functional description

These sections describe the functionality of the various building blocks of the USB.

44.5.1 USB dual role device/host controller

The USB Dual Role controller is an EHCI-compatible USB core which supports High Speed / Full Speed / Low Speed operation.

The USB controller core supports HS/FS/LS operation in Host mode and HS/FS operation in device mode.

44.5.1.1 Host mode

The controller supports direct connection of a HS/FS/LS device with on-chip UTMI transceiver. Although there is no separate Transaction Translator block in the system, the transaction translator function normally associated with a USB 2.0 high speed hub has been implemented within the DMA and protocol engine blocks to support connection to full and low speed devices.

44.5.1.2 Peripheral (Device) Mode

- Up to 6 bidirectional endpoints
- High/Full speed operation
- Remote wake-up capable

44.5.2 USB Power Control Block

The USB controller supports low-power suspend and wake-up functionality. The power control block allows for placing the transceiver in low power mode when USB bus is IDLE, and supports local and remote wake-up to bring the transceiver out of low power mode when needed. Additionally, the power control block can wake-up the ARM platform from sleep mode by generating an interrupt.

44.5.2.1 Entering Low Power Suspend mode

Low Power Suspend mode is always entered under the control of the driver software by setting the appropriate bit in USB core register PORTSC1. Once the controller is suspended, the clocks to the USB can be stopped.

44.5.2.2 Wake-up events

The power control block monitors the USB bus when the USB core is in the suspend state. Depending on whether the core is on Host or Device mode, a number of wake-up conditions are monitored. Upon detection of a wake-up condition, an interrupt (asynchronous) can be generated to ARM platform if related wake-up interrupt enable bit is set. This interrupt also re-activates the ARM platform clocks if they were stopped during the suspend.

44.5.2.2.1 Host mode events

The Host controller wakes-up on the following events:

- Remote wake-up request

A peripheral can request the host to re-activate the bus by driving wake-up signaling on the DM/DP lines. The power control block sends a wake-up request to the USB core when J-K transition on DM/DP line is detected.

- Wake-up on overcurrent

If wake-up on overcurrent is enabled (WKOC bit in USB core register PORTSC1 is set to 1), the power control block sends a wake-up request to the USB core when over-current event is detected.

- Wake-up on disconnect

If wake-up on disconnection is enabled (WKDC bit in USB core register PORTSC1 is set to 1), the power control block sends a wake-up request to the USB core when disconnection event is detected (J-SE0/K-SE0 transition on DM/DP line).

- Wake-up on connect

If wake-up on connection is enabled (WKCN bit in USB core register PORTSC1 is set to 1), the power control sub-block sends a wake-up request to the USB core when connection event is detected (SE0-J/SE0-K transition on DM/DP line).

For a detailed description of register bits WKOC, WKDC, and WKCN, please see Port Status & Control (USB_PORTSC1).

44.5.2.2.2 Device mode events

When the OTG controller is configured for peripheral operation, the power control block sends a wake-up request to the USB core when any non-IDLE state is detected on the USB bus.

NOTE

Wake Up from USB needs following configuration:

1. REFTOP inside ANADIG must be ON in stop mode (required for Device Mode).
   - If ANATOP_STOP_MODE is used for stop mode (ANATOP_STOP_MODE bit is set in CCM Low Power Control Register), then 'stop_mode_config bit' in “ANADIG_ANA_MISC0” register inside ANADIG must be set to ‘1’.
   - If ANATOP_STOP_MODE is NOT used for stop mode, then keep “reftop_pwd” bit to ‘0’ in “anadig_anamisc0” register inside ANADIG.
2. Either Main Regulator OR Weak Regulator (1P1 & 2P5) must be ON in stop mode.

- If ANATOP_STOP_MODE is used for stop mode (ANATOP_STOP_MODE bit is set in CCM Low Power Control Register), then “enable_weak_linreg” bit in “anadig_reg2p5” register inside ANADIG must be set to ‘1’.
- If ANATOP_STOP_MODE is NOT used for stop mode, then either enable Weak Regulators 1P1 & 2P5 (set “enable_weaklinreg” bit in “anadig_reg2p5” register inside ANADIG) or enable Main regulators 1P1 & 2P5 (set “enable_linreg” bit in “anadig_reg1p1” & “anadig_reg1p1” registers inside ANADIG). To reduce power consumption in stop mode, main regulators are turned off, and weak regulators are turned on to support wake-up from USB.

44.5.3 Interrupts

44.5.3.1 USB core interrupts

Each USB core uses one dedicated vector in the interrupt table. The vector numbers associated with each of the cores can be found in the Interrupt section.

With the exception of the wake-up interrupts, all of the interrupt sources are controlled in the USB cores. Refer to the Interrupt Enable Register (USB_USBINTR) for details.

44.5.3.2 USB wake-up interrupts

Each USB core has an associated wake-up interrupt. The wake-up interrupts are generated outside of the USB controller cores, but use the same vector as the corresponding USB controller cores' interrupt. These interrupts are generated by the Power Control blocks, which run on the 32 KHz standby clock.The wake-up interrupt is designed to work even when the USB and ARM platform clocks are disabled, such that a wake-up condition on the USB bus can reactivate the ARM platform clocks.

Because the wake-up interrupt is generated and cleared on a 32 KHz clock, this interrupt request responds very slowly to clear actions. For this reason, the software must disable the wake-up interrupt to clear the request flag. Disabling the interrupt masks the request instantaneously as this is clocked by the ARM platform clock. The software should than wait for at least three 32 KHz clock cycles before re-enabling this interrupt to allow

sufficient time for the request flag to clear. Because this interrupt is only used during low power modes of the USB, it is sufficient to enable the wake-up interrupt just prior to entering the USB suspend mode.

44.6 USB operation model

This section describes the detailed application knowledge for OTG ports. It can be generally divided in two parts, one for Host and the other for Device. The Host part applies to the OTG ports when operating in Host mode. The Device part applies to the OTG ports when operating in Device mode.

44.6.1 Register interface

Slave accesses from the controlling processor enables access to the configuration, control, and status registers. One function of the system address map is the registers base address, which must begin on a 32-bit boundary. Register offset definitions are listed in the table below.

Configuration, control and status registers are divided into three categories: identification, capability, and operational registers.

- Identification registers are used to declare the slave interface presence along with the complete set of the hardware configuration parameters.
- Static, read-only capability registers define the software limits, restrictions, and capabilities of the host/device controller.
- Operational registers are comprised of dynamic control or status registers that may be read only, read/write, or read/write to clear. The following sections define the use of these registers.

EHCI registers are listed alongside device registers to show the complementary nature of host and device control.

Table 44-145 describes the interface register sets.

Table 44-145. Interface register sets

Offset	Register Set	Explanation
000h-07Ch	Identification registers	Identification registers are used to declare the slave interface presence and include a table of the hardware configuration parameters.
100h-124h	Capability registers	Capability registers specify the limits, restrictions, and capabilities of a host/ device controller implementation. These values are used as parameters to the host/device controller driver.

Table continues on the next page...

Table 44-145. Interface register sets (continued)

080h-0FCh 140h-1FCh	Operational registers	Operational registers are used by the system software to control and monitor the operational state of the host/device controller.

44.6.1.1 Configuration, Control and Status Register Set

Table 44-146 describes the Device/Host capability registers.

Table 44-146. Device/Host capability registers

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
000h	4	USB_ID	Identification Register	O	O
004h	4	USB_HWGENERAL	General Hardware Parameters	O	O
008h	4	USB_HWHOST	Host Hardware Parameters	X	O
00Ch	4	USB_HWDEVICE	Device Hardware Parameters	O	X
010h	4	USB_HWTXBUF	TX Buffer Hardware Parameters	O	O
014h	4	USB_HWRXBUF	RX Buffer Hardware Parameters	O	O
018-07Fh		-	Reserved		
080h	4	USB_GPTIMER0LD	General Purpose Timer #0 Load Register	O	O
084h	4	USB_GPTIMER0CTRL	General Purpose Timer #0 Control Register	O	O
088h	4	USB_GPTIMER1LD	General Purpose Timer #1 Load Register	O	O
08Ch	4	USB_GPTIMER1CTRL	General Purpose Timer #1 Control Register	O	O
090h	4	USB_SBUSCFG	System Bus Interface Configuration Register	O	O
094-09Fh		-	Reserved		
100h	1	USB_CAPLENGTH	Capability Register Length	O	O
101h		-	Reserved		
102h	2	USB_HCIVERSION	Host Controller Interface Version Number	X	O
104h	4	USB_HCSPARAMS	Host Controller Structural Parameters	X	O
108h	4	USB_HCCPARAMS	Host Controller Capability Parameters	X	O
10C-11Fh		-	Reserved		
120h	2	USB_DCIVERSION	Device Controller Interface Version Number	O	X
122h	2	-	Reserved		
124h	4	USB_DCCPARAMS	Device Controller Capability Parameters	O	X
128-13Fh		-	Reserved		
140h	4	USB_USBCMD	USB Command Register	O	O
144h	4	USB_USBSTS	USB Status Register	O	O
148h	4	USB_USBINTR	USB Interrupt Enable Register	O	O
14Ch	4	USB_FRINDEX	USB Frame Index	O	O
150h	4	-	Reserved		

Table continues on the next page...

Table 44-146. Device/Host capability registers (continued)

Offset	Size (Bytes)	Mnemonic	Register Name	Device Mode	Host Mode
154h	4	USB_PERIODICLISTBASE	Frame List Base Address	X	O
		USB_DEVICEADDR	USB Device Address	O	X
158h	4	USB_ASYNCLISTADDR	Next Asynchronous List Address	X	O
		USB_ENDPOINTLISTADDR	Address at Endpoint list in memory	O	X
15Ch	4	-	Reserved		
160h	4	USB_BURSTSIZE	Programmable Burst Size	O	O
164h	4	USB_TXFILLTUNING	Host Transmit Pre-Buffer Packet Tuning	X	O
168h	4	-	Reserved		
16Ch	4	USB_IC_USB	IC_USB enable and voltage negotiation	O	O
170h	4	-	Reserved		
174h	4	USB_ENDPTNAK	Endpoint NAK register	O	X
178h	4	USB_ENDPTNAKEN	Endpoint NAK Enable register	O	X
17Ch	4	-	Reserved		
180h	4	USB_CONFIGFLAG	Configured Flag Register	X	O
184h	4	USB_PORTSC1	Port Status/Control Register 1	O	O
188-1A3h		-	Reserved		
1A4h	4	USB_OTGSC	On-The-Go Status/Control Register	O	O
1A8h	4	USB_USBMODE	USB Device Mode	O	O
1ACh	4	USB_ENDPTSETUPSTAT	Endpoint Setup Status	O	X
1B0h	4	USB_ENDPTPRIME	Endpoint Initialization	O	X
1B4h	4	USB_ENDPTFLUSH	Endpoint De-Initialization	O	X
1B8h	4	USB_ENDPTSTATUS	Endpoint Status	O	X
1BCh	4	USB_ENDPTCOMPLETE	Endpoint Complete	O	X
1C0 1C4 ... 1D4h	24	USB_ENDPTCTRL0 USB_ENDPTCTRL1 .... USB_ENDPTCTRL5	Endpoint Control Register 0 - 5	O	X

44.6.1.2 Identification registers

Identification registers are used to declare the slave interface presence and include a table of the hardware configuration parameters.

44.6.2 Host data structures

This section defines the interface data structures used to communicate control, status, and data between HCD (software) and the Enhanced Host Controller (hardware). The data structure definitions in this chapter support a 32-bit memory buffer address space. The interface consists of a Periodic Schedule, Periodic Frame List, Asynchronous Schedule, Isochronous Transaction Descriptors, Split-transaction Isochronous Transfer Descriptors, Queue Heads, and Queue Element Transfer Descriptors.

The periodic frame list is the root of all periodic (isochronous and interrupt transfer type) support for the host controller interface. The asynchronous list is the root for all the bulk and control transfer type support. Isochronous data streams are managed using Isochronous Transaction Descriptors. Isochronous split-transaction data streams are managed with Split-transaction Isochronous Transfer Descriptors. All Interrupt, Control, and Bulk data streams are managed via queue heads and Queue Element Transfer Descriptors. These data structures are optimized to reduce the total memory footprint of the schedule and to reduce (on average) the number of memory accesses needed to execute a USB transaction.

Note that software must ensure that no interface data structure reachable by the EHCI host controller spans a 4 K-page boundary.

The data structures defined in this section are (from the host controller's perspective) a mix of read-only and read/writeable fields. The host controller must preserve the read-only fields on all data structure writes.

44.6.2.1 Periodic Frame List

This schedule is for all periodic transfers (isochronous and interrupt). The periodic schedule is referenced from the operational registers space using the USB_PERIODICLISTBASE address register and the USB_FRINDEX register. The periodic schedule is based on an array of pointers called the Periodic Frame List. The USB_PERIODICLISTBASE address register is combined with the USB_FRINDEX register to produce a memory pointer into the frame list. The Periodic Frame List implements a sliding window of work over time.

Figure 44-140 shows the organization of periodic schedule.

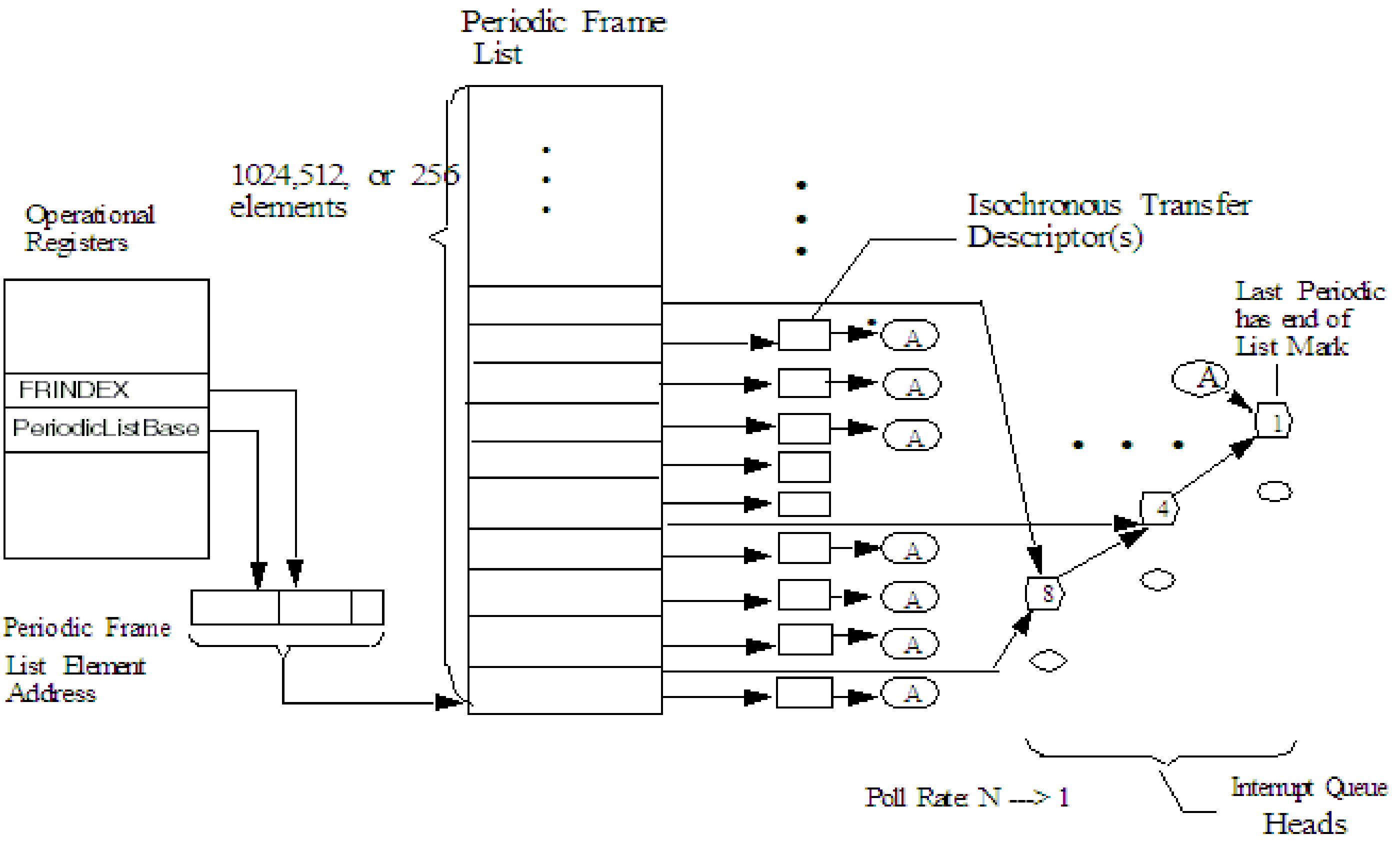


Figure 44-140. Periodic schedule organization

Split transaction Interrupt, Bulk and Control are also managed using queue heads and queue element transfer descriptors.

The periodic frame list is a 4 KB page-aligned array of Frame List Link pointers. The length of the frame list may be programmable. The programmability of the periodic frame list is exported to system software via the USB_HCCPARAMS register. If non-programmable, the length is 1024 elements. If programmable, the length can be selected by system software as one of 256, 512, or 1024 elements. An implementation must support all three sizes. Programming the size (that is, the number of elements) is accomplished by system software writing the appropriate value into Frame List Size field in the USB_USBCMD register.

Frame List Link pointers direct the host controller to the first work item in the frame's periodic schedule for the current micro-frame. The link pointers are aligned on DWord boundaries within the Frame List.

Table 44-147 illustrates the format of the Frame list element pointer.

Table 44-147. Format of Frame List Element Pointer

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Frame List Link Pointer																											0		Typ			03-00H

Frame List Link pointers always reference memory objects that are 32-byte aligned. The referenced object may be an isochronous transfer descriptor for high-speed devices, a split-transaction isochronous transfer descriptor (for full-speed isochronous endpoints), or a queue head (used to support high-, full- and low-speed interrupt). System software should not place non-periodic schedule items into the periodic schedule. The least significant bits in a frame list pointer are used to key the host controller as to the type of object the pointer is referencing.

The least significant bit is the T-Bit (bit 0). When this bit is set to a one, the host controller never uses the value of the frame list pointer as a physical memory pointer. The Typ field is used to indicate the exact type of data structure being referenced by this pointer. The value encodings are.

Table 44-148. Typ field value definitions

Value	Meaning
00b	Isochronous Transfer Descriptor
01b	Queue Head
10b	Split Transaction Isochronous Transfer Descriptor
11b	Frame Span Traversal Node

44.6.2.2 Asynchronous List Queue Head Pointer

The Asynchronous Transfer List (based at the USB_ASYNCLISTADDR register) is where all the control and bulk transfers are managed. Host controllers use this list only when it reaches the end of the periodic list, the periodic list is disabled, or the periodic list is empty.

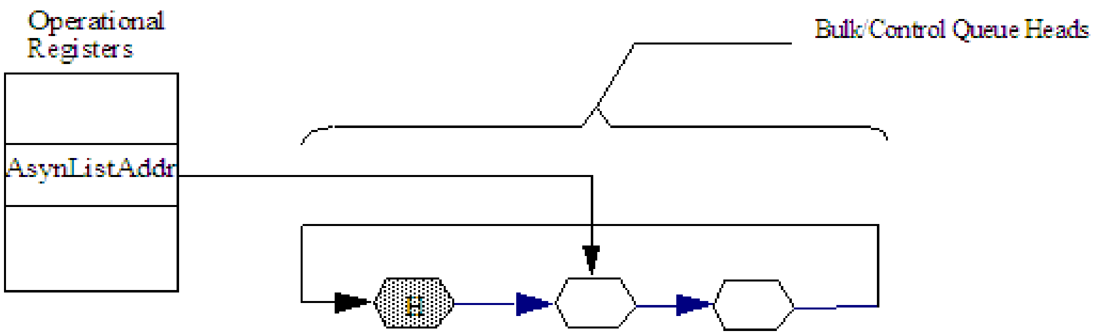


Figure 44-141. Asynchronous schedule organization

The Asynchronous list is a simple circular list of queue heads. The USB_ASYNCLISTADDR register is simply pointer to the next queue head. This implements a pure round-robin service for all queue heads linked into the asynchronous list.

44.6.2.3 Isochronous (High-Speed) Transfer Descriptor (iTD)

The format of an isochronous transfer descriptor is shown in Table 44-149. This structure is used only for high-speed isochronous endpoints. All other transfer types should use queue structures. Isochronous TDs must be aligned on a 32-byte boundary.

Table 44-149. Isochronous Transaction Descriptor (iTD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Next Link Pointer																											0		Typ		T	03-00H
Status				Transaction 0 Length												ioc	PG*			Transaction 0 Offset*												07-04H
Status				Transaction 1 Length												ioc	PG*			Transaction 1 Offset*												0B-08H
Status				Transaction 2 Length												ioc	PG*			Transaction 2 Offset*												0F-0CH
Status				Transaction 3 Length												ioc	PG*			Transaction 3 Offset*												13-10H
Status				Transaction 4 Length												ioc	PG*			Transaction 4 Offset*												17-14H
Status				Transaction 5 Length												ioc	PG*			Transaction 5 Offset*												1B-18H
Status				Transaction 6 Length												ioc	PG*			Transaction 6 Offset*												1F-1CH

Table continues on the next page...

Table 44-149. Isochronous Transaction Descriptor (iTD) (continued)

Status	Transaction 7 Length	ioc	PG*	Transaction 7 Offset*	23-20H
Buffer Pointer (Page 0)		EndPt	R	Device Address	27-24H
Buffer Pointer (Page 1)		I/O	Maximum Packet Size		2B-28H
Buffer Pointer (Page 2)		-		Mult	2F-2CH
Buffer Pointer (Page 3)		-			33-30H
Buffer Pointer (Page 4)		-			37-34H
Buffer Pointer (Page 5)		-			3B-38H
Buffer Pointer (Page 6)		-			3F-3CH

(shaded)	Host Controller Read/Write	(unshaded)	Host Controller Read Only.

These fields may be modified by the host controller if the I/O field indicates an OUT.

44.6.2.3.1 Next Link Pointer-Host Data Structures

The first DWord of an iTD is a pointer to the next schedule data structure. Table 44-150 describes the Next Schedule Element pointer field.

Table 44-150. Next Schedule Element pointer

Bit	Description
31-5	Link Pointer (LP). These bits correspond to memory address signals [31:5], respectively. This field points to another Isochronous Transaction Descriptor (iTD/siTD) or Queue Head (QH).
4-3	Reserved. These bits are reserved and their value has no effect on operation. Software should initialize this field to zero.
2-1	QH/(s)iTD Select (Typ). This field indicates to the Host Controller whether the item referenced is an iTD, siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor 11b FSTN (frame span traversal node)
0	Terminate (T). 1= Link Pointer field is not valid. 0= Link Pointer field is valid.

44.6.2.3.2 iTD Transaction Status and Control List

DWords 1 through 8 are eight slots of transaction control and status. Each transaction description includes:

- Status results field
- Transaction length (bytes to send for OUT transactions and bytes received for IN transactions).
- Buffer offset. The PG and Transaction X Offset fields are used with the buffer pointer list to construct the starting buffer address for the transaction.

The host controller uses the information in each transaction description plus the endpoint information contained in the first three DWords of the Buffer Page Pointer list, to execute a transaction on the

Table 44-151 describes iTD Transaction Status and Control fields.

Table 44-151. iTD Transaction Status and Control

Bit	Description	
31-28	Status. This field records the status of the transaction executed by the host controller for this slot. This field is a bit vector with the following encoding:	
	Bit	Definition
	31	Active. Set to one by software to enable the execution of an isochronous transaction by the Host Controller. When the transaction associated with this descriptor is completed, the Host Controller sets this bit to zero indicating that a transaction for this element should not be executed when it is next encountered in the schedule.
	30	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). If an overrun condition occurs, no action is necessary.
	29	Babble Detected. Set to one by the Host Controller during status update when" babble" is detected during the transaction generated by this descriptor.
	28	Transaction Error (XactErr). Set to one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). This bit may only be set for isochronous IN transactions.
27-16	Transaction X Length. For an OUT, this field is the number of data bytes the host controller sends during the transaction. The host controller is not required to update this field to reflect the actual number of bytes transferred during the transfer. For an IN, the initial value of the endpoint to deliver. During the status update, the host controller writes back the field is the number of bytes the host expects the number of bytes successfully received. The value in this register is the actual byte count (0‡zero length data, 1‡one byte, 2‡two bytes, etc.). The maximum value this field may contain is 0xC00 (3072).	
15	Interrupt On Complete (IOC). If this bit is set to one, it specifies that when this transaction completes, the Host Controller should issue an interrupt at the next interrupt threshold.	
14-12	Page Select (PG). These bits are set by software to indicate which of the buffer page pointers the offset field in this slot should be concatenated to produce the starting memory address for this transaction. The valid range of values for this field is 0 to 6.	
11-0	Transaction X Offset. This field is a value that is an offset, expressed in bytes, from the beginning of a buffer. This field is concatenated onto the buffer page pointer indicated in the adjacent *PG* field to produce the starting buffer address for this transaction.	

44.6.2.3.3 iTD Buffer Page Pointer List (Plus)

DWords 9-15 of an isochronous transaction descriptor are nominally page pointers (4 K aligned) to the data buffer for this transfer descriptor. This data structure requires the associated data buffer to be contiguous (relative to virtual memory), but allows the physical memory pages to be non-contiguous. Seven page pointers are provided to support the expression of eight isochronous transfers. The seven pointers allow for 3 (transactions) * 1024 (maximum packet size) * 8 (transaction records) (24576 bytes) to be moved with this data structure, regardless of the alignment offset of the first page.

Because each pointer is a 4 K aligned page pointer, the least significant 12 bits in several of the page pointers are used for other purposes.

Tables below illustrate the field descriptions.

Table 44-152. iTD Buffer Pointer Page 0 (Plus)

Bit	Description
31-12	Buffer Pointer (Page 0). This is a 4 K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-8	Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.
7	Reserved. Bit reserved for future use and should be initialized by software to zero.
6-0	Device Address. This field selects the specific device serving as the data source or sink.

Table 44-153. iTD Buffer Pointer Page 1 (Plus)

Bit	Description
31-12	Buffer Pointer (Page 1). This is a 4K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11	Direction (I/O). 0 = OUT; 1 = IN. This field encodes whether the high-speed transaction should use an IN or OUT PID.
10-0	Maximum Packet Size. This directly corresponds to the maximum packet size of the associated endpoint *(wMaxPacketSize)*. This field is used for high-bandwidth endpoints where more than one transaction is issued per transaction description (per micro-frame). This field is used with the *Multi* field to support high-bandwidth pipes. This field is also used for all IN transfers to detect packet babble. Software should not set a value larger than 1024 (400h). Any value larger yields undefined results.

Table 44-154. iTD Buffer Pointer Page 2 (Plus)

Bit	Description
31-12	Buffer Pointer. This is a 4K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-2	Reserved. This bit reserved for future use and should be set to zero.

Table continues on the next page...

Table 44-154. iTD Buffer Pointer Page 2 (Plus) (continued)

1-0	Multi. This field is used to indicate to the host controller the number of transactions that should be executed per transaction description (per micro-frame). The valid values are: Value Meaning 00b Reserved. A zero in this field yields undefined results. 01b One transaction to be issued for this endpoint per micro- frame 10b Two transactions to be issued for this endpoint per micro- frame 11b Three transactions to be issued for this endpoint per micro- frame

Table 44-155. iTD Buffer Pointer Page 3-6

Bit	Description
31-12	Buffer Pointer. This is a 4 K aligned pointer to physical memory. Corresponds to memory address bits [31:12].
11-0	Reserved. These bits reserved for future use and should be set to zero.

44.6.2.4 Split Transaction Isochronous Transfer Descriptor (siTD)

All Full-speed isochronous transfers through the internal transaction translator are managed using the siTD data structure. This data structure satisfies the operational requirements for managing the split transaction protocol.

Figure below shows the Split Transaction Isochronous Transfer Descriptor (siTD).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td colspan="27">Next Link Pointer</td><td colspan="2">0</td><td colspan="2">Typ</td><td>T</td><td>03-00H</td></tr>
<tr><td>I/O</td><td colspan="7">Port Number</td><td>-</td><td colspan="7">Hub Addr</td><td colspan="4">-</td><td colspan="4">EndPt</td><td>-</td><td colspan="7">Device Address</td><td>07-04H</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="8">µFrame C-mask</td><td colspan="8">µFrame S-mask</td><td>0B-08H</td></tr>
<tr><td>ioc</td><td>P</td><td colspan="4">-</td><td colspan="10">Total Bytes to Transfer</td><td colspan="8">µFrame C-prog-mask</td><td colspan="8">Status</td><td>0F-0CH</td></tr>
<tr><td colspan="20">Buffer Pointer (Page 0)</td><td colspan="12">Current Offset</td><td>13-10H</td></tr>
<tr><td colspan="20">Buffer Pointer (Page 1)</td><td colspan="7">-</td><td colspan="2">TP</td><td colspan="3">T-count</td><td>17-14H</td></tr>
<tr><td colspan="27">Back Pointer</td><td colspan="4">0</td><td>T</td><td>1B-18H</td></tr>
</table>

Table 44-156. Split-transaction Isochronous Transaction Descriptor (siTD)

	Host Controller Read/Write		Host Controller Read Only.

44.6.2.4.1 Next Link Pointer

DWord0 of a siTD is a pointer to the next schedule data structure.

Table 44-157 describes the Next Link Pointer fields.

Table 44-157. Next Link Pointer

Bit	Description
31-5	Next Link Pointer (LP). This field contains the address of the next data object to be processed in the periodic list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved. These bits must be written as zeros.
2-1	QH/(s)iTD Select (Typ). This field indicates to the Host Controller whether the item referenced is an iTD/siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor 11b FSTN (frame span traversal node)
0	Terminate (T). 1=Link Pointer field is not valid. 0=Link Pointer is valid.

44.6.2.4.2 siTD Endpoint Capabilities/Characteristics

DWords 1 and 2 specify static information about the full-speed endpoint, the addressing of the parent Companion Controller, and micro-frame scheduling control.

Tables below describe the Endpoint and transaction translator characteristics and micro-frame schedule control fields.

Table 44-158. Endpoint and Transaction Translator Characteristics

Bit	Description
31	Direction (I/O).0 = OUT; 1 = IN. This field encodes whether the full-speed transaction should be an IN or OUT.
30-24	Port Number. This field is the port number of the recipient Transaction Translator.
23	Reserved. Bit reserved and should be set to zero.
22-16	Hub Address. This field holds the device address of the Companion Controllers' hub.
15-12	Reserved. Field reserved and should be set to zero.
11-8	Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.
7	Reserved. Bit is reserved for future use. It should be set to zero.
6-0	Device Address. This field selects the specific device serving as the data source or sink.

Table 44-159. Micro-frame Schedule Control

Bit	Description
31-16	Reserved. This field reserved for future use. It should be set to zero.

Table continues on the next page...

Table 44-159. Micro-frame Schedule Control (continued)

15-8	Split Completion Mask (mFrame C-Mask). This field (along with the *Active* and *SplitX- state* fields in the *Status* byte) is used to determine during which micro-frames the host controller should execute complete-split transactions. When the criteria for using this field is met, an all zeros value has undefined behavior. The host controller uses the value of the three low-order bits of the FRINDEX register to index into this bit field. If the FRINDEX register value indexes to a position where the m*Frame C-Mask* field is a one, then this siTD is a candidate for transaction execution. There may be more than one bit in this mask set.
7-0	Split Start Mask (mFrame S-mask). This field (along with the *Active* and *SplitX-state* fields in the *Status* byte) is used to determine during which micro-frames the host controller should execute start-split transactions. The host controller uses the value of the three low-order bits of the FRINDEX register to index into this bit field. If the FRINDEX register value indexes to a position where the m*Frame S-mask* field is a one, then this siTD is a candidate for transaction execution. An all zeros value in this field, in combination with existing periodic frame list has undefined results.

44.6.2.4.3 siTD Transfer State

DWords 3-6 are used to manage the state of the transfer. Table 44-160 describes siTD transfer state fields.

Table 44-160. siTD Transfer Status and Control

Bit	Description	
31	Interrupt On Complete (ioc). 0 = Do not interrupt when transaction is complete. 1 = Do interrupt when transaction is complete. When the host controller determines that the split transaction has completed it asserts a hardware interrupt at the next interrupt threshold.	
30	Page Select (P). Used to indicate which data page pointer should be concatenated with the *CurrentOffset* field to construct a data buffer pointer (0 selects *Page 0* pointer and 1 selects *Page 1*). The host controller is not required to write this field back when the siTD is retired (*Active* bit transitioned from a one to a zero).	
29-26	Reserved. This field reserved for future use and should be set to zero.	
25-16	Total Bytes To Transfer. This field is initialized by software to the total number of bytes expected in this transfer. Maximum value is 1023 (3FFh)	
15-8	μFrame Complete-split Progress Mask (C-prog-Mask). This field is used by the host controller to record which split-completes has been executed.	
7-0	Status. This field records the status of the transaction executed by the host controller for this slot. This field is a bit vector with the following encoding:	
	Bit	Definition
	7	Active. Set to one by software to enable the execution of an isochronous split transaction by the Host Controller.
	6	ERR. Set to a one by the Host Controller when an ERR response is received from the Companion Controller.
	5	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). In the case of an under run, the Host Controller transmits an incorrect CRC (thus invalidating the data at the endpoint). If an overrun condition occurs, no action is necessary.
	4	Babble Detected. Set to a one by the Host Controller during status update when" babble" is detected during the transaction generated by this descriptor.

Table continues on the next page...

Table 44-160. siTD Transfer Status and Control (continued)

Bit	Description	
	3	Transaction Error (XactErr). Set to a one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). This bit is set only for IN transactions.
	2	Missed Micro-Frame. The host controller detected that a host-induced hold- off caused the host controller to miss a required complete-split transaction.
	1	Split Transaction State (SplitXstate). The bit encodings are: Value Meaning 00b Do Start Split. This value directs the host controller to issue a Start split transaction to the endpoint when a match is encountered in the S-mask. 01b Do Complete Split. This value directs the host controller to issue a Complete split transaction to the endpoint when a match is encountered in the C-mask.
	0	Reserved. Bit reserved for future use and should be set to zero.

44.6.2.4.4 siTD Buffer Pointer List (plus)

DWords 4 and 5 are the data buffer page pointers for the transfer. This structure supports one physical page cross. The most significant 20 bits of each DWord in this section are the 4 K (page) aligned buffer pointers.

The least significant 12 bits of each DWord are used as additional transfer state. Table 44-161 describes the siTD buffer pointer fields.

Table 44-161. Buffer Page Pointer List (plus)

Bit	Description	
31-12	Buffer Pointer List. Bits [31:12] of DWords 4 and 5 are 4 K paged aligned, physical memory addresses. These bits correspond to physical address bits [31:12] respectively. The lower 12 bits in each pointer are defined and used as specified below. The field *P* specifies the *current* active pointer	
11-0	Page 0: Current Offset. The 12 least significant bits of the Page 0 pointer is the current byte offset for the current page pointer (as selected with the page indicator bit (*P* field)). The host controller is not required to write this field back when the siTD is retired (*Active* bit transitioned from a one to a zero). The least significant bits of Page 1 pointer is split into three sub-fields Page 1:	
	Bits	Description
	11-5	Reserved

Table continues on the next page...

Table 44-161. Buffer Page Pointer List (plus) (continued)

Bit	Description	
	4-3	Transaction position (TP). This field is used with T-count to determine whether to send *all*, *first*, *middle*, or *last* with each outbound transaction payload. System software must initialize this field with the appropriate starting value. The host controller must correctly manage this state during the lifetime of the transfer. The bit encodings are: Value Meaning 00b All. The entire full-speed transaction data payload is in this transaction (that is, less than or equal to 188 bytes). 01b Begin. This is the first data payload for a full- speed that is greater than 188 bytes.transaction 10B Mid. This is the *middle* payload for a full-speed OUT transaction that is larger than 188 bytes. 11b End. This is the *last* payload for a full-speed OUT transaction that was larger than 188 bytes.
	2-0	Transaction count (T-Count). Software initializes this field with the number of OUT start-splits this transfer requires. Any value larger than 6 is undefined.

44.6.2.4.5 siTD Back Link Pointer

DWord 6 of a siTD is simply another schedule link pointer. This pointer is always zero, or references a siTD. This pointer cannot reference any other schedule data structure.

Table 44-162 describes the siTD back link pointer fields

.

Table 44-162. siTD Back Link Pointer

Bit	Description
31-5	siTD Back Pointer. This field is a physical memory pointer to a siTD.
4-1	Reserved. This field is reserved for future use. It should be set to zero.
0	Terminate (T). 1 = siTD Back Pointer field is not valid. 0 = siTD Back Pointer field is valid.

44.6.2.5 Queue Element Transfer Descriptor (qTD)

This data structure is only used with a queue head. This data structure is used for one or more USB transactions. This data structure is used to transfer up to 20480 (5*4096) bytes. The structure contains two structure pointers used for queue advancement, a DWord of transfer state, and a five-element array of data buffer pointers. This structure is 32 bytes (or one 32-byte cache line). This data structure must be physically contiguous.

The buffer associated with this transfer must be virtually contiguous. The buffer may start on any byte boundary. A separate buffer pointer list element must be used for each physical page in the buffer, regardless of whether the buffer is physically contiguous.

Host controller updates (host controller writes) to stand-alone qTDs only occur during transfer retirement. References in the following field definitions of updates to the qTD are to the qTD portion of a queue head.

Table 44-163 shows the Queue head data structure.

Table 44-163. Queue Head Data Structure

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Next qTD Pointer																											0				T	03-00H
Alternate Next qTD Pointer																											0				T	07-04H
dt	Total Bytes to Transfer															ioc	C_Page			Cerr		PID Code		Status								0B-08H
Buffer Pointer (page 0)																				Current Offset												0F-0CH
Buffer Pointer (page 0)																				-												13-10H
Buffer Pointer (page 0)																				-												17-14H
Buffer Pointer (page 0)																				-												1B-18H
Buffer Pointer (page 0)																				-												1F-1CH

(shaded)	Host Controller Read/Write		Host Controller Read Only.

Queue Element Transfer Descriptors must be aligned on 32-byte boundaries.

44.6.2.5.1 Next qTD Pointer

The first DWord of an element transfer descriptor is a pointer to another transfer element descriptor. Table 44-164 describes Next qTD pointer fields.

Table 44-164. qTD Next Element Transfer Pointer (DWord 0)

Bit	Description
31-5	Next Transfer Element Pointer. This field contains the physical memory address of the next qTD to be processed. The field corresponds to memory address signals[31:5], respectively.
4-1	Reserved
0	Terminate (T). 1= pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Host Controller that there are no more valid entries in the queue.

44.6.2.5.2 Alternate Next qTD Pointer

The second DWord of a queue element transfer descriptor is used to support hardware-only advance of the data stream to the next client buffer on short packet. To be more explicit the host controller always uses this pointer when the current qTD is retired due to short packet. Table 44-165 describes the TD Alternate Next Element Transfer Pointer field descriptions.

Table 44-165. TD Alternate Next Element Transfer Pointer (DWord 1)

Bit	Description
31-5	Alternate Next Transfer Element Pointer. This field contains the physical memory address of the next qTD to be processed in the event that the current qTD execution encounters a short packet (for an IN transaction). The field corresponds to memory address signals [31:5], respectively.
4-1	Reserved
0	Terminate (T). 1= pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Host Controller that there are no more valid entries in the queue.

44.6.2.5.3 qTD Token

The third DWord of a queue element transfer descriptor contains most of the information the host controller requires to execute a USB transaction (the remaining endpoint-addressing information is specified in the queue head).

NOTE

The field descriptions forward reference fields defined in the queue head. Where necessary, these forward references are preceded with a QH notation.

Table 44-166 describes the TD Token fields.

Table 44-166. TD Token (DWord 2)

Bit	Description	
31	Data Toggle. This is the data toggle sequence bit. The use of this bit depends on the setting of the *Data Toggle Control* bit in the queue head.	
30-16	Total Bytes to Transfer. This field specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes actually moved during the transaction, only on the successful completion of the transaction. The maximum value software may store in this field is 5 * 4K (5000H). This is the maximum number of bytes 5 page pointers can access. If the value of this field is zero when the host controller fetches this transfer descriptor (and the active bit is set), the host controller executes a zero-length transaction and retires the transfer descriptor. It is not a requirement for OUT transfers that *Total Bytes To Transfer* be an even multiple of QHD.Maximum Packet Length. If software builds such a transfer descriptor for an OUT transfer, the last transaction is always less than QHD.Maximum Packet Length. Although it is possible to create a transfer up to 20K this assumes the 1^{st} offset into the first page is 0. When the offset cannot be predetermined, crossing past the 5th page can be guaranteed by limiting the total bytes to 16K**. Therefore, the maximum recommended transfer is 16 K(4000H).	
15	Interrupt On Complete (IOC). If this bit is set to a one, it specifies that when this qTD is completed, the Host Controller should issue an interrupt at the next interrupt threshold.	
14-12	Current Page (C_Page). This field is used as an index into the qTD buffer pointer list. Valid values are in the range 0H to 4H. The host controller is not required to write this field back when the qTD is retired.	
11-10	Error Counter (CERR). This field is a 2-bit down counter that keeps track of the number of consecutive Errors detected while executing this qTD. If this field is programmed with a non-zero value during set-up, the Host Controller decrements the count and writes it back to the qTD if the transaction fails. If the counter counts from one to zero, the Host Controller marks the qTD inactive, sets the *Halted* bit to a one, and error status bit for the error that caused *CERR* to decrement to zero. An interrupt is generated if the *USB Error Interrupt Enable* bit in the USBINTR register is set to a one. If HCD programs this field to zero during set-up, the Host Controller does not count errors for this qTD and there is no limit on the retries of this qTD. Note that write-backs of intermediate execution state are to the queue head overlay area, not the qTD. Error Decrement Counter Transaction Error Yes Data Buffer Error No[3] Stalled No[1] Babble Detected No[1] No Error No[2]	
	Error	Decrement Counter Error Decrement Counter
	1	Detection of Babble or Stall automatically halts the queue head. Thus, count is not decremented
	2	If the EPS field indicates a HS device or the queue head is in the Asynchronous Schedule (and *PIDCode* indicates an IN or OUT) and a bus transaction completes and the host controller does not detect a transaction error, then the host controller should reset *CERR* to extend the total number of errors for this transaction. For example, *CERR* should be reset with maximum value (3) on each successful completion of a transaction. The host controller must never reset this field if the value at the start of the transaction is 00b. See Split Transaction Interrupt for CERR adjustment rules when the EPS field indicates a FS or LS device and the queue head is in the Periodic Schedule. See Asynchronous - Do Complete Split for CERR adjustment rules when the EPS field indicates a FS or LS device, the queue head is in the Asynchronous schedule and the *PIDCode* indicates a SETUP.
	3	Data buffer errors are host problems. They don't count against the device's retries.
	NOTE: Software must not program CERR to a value of zero when the EPS field is programmed with a value indicating a Full- or Low-speed device. This combination could result in undefined behavior.	
9-8	PID Code. This field is an encoding of the token, which should be used for transactions associated with this transfer descriptor. Encodings are:	
	00b	OUT Token generates token (E1H)
	01b	IN Token generates token (69H)

Table continues on the next page...

Table 44-166. TD Token (DWord 2) (continued)

Bit	Description	
	10b	SETUP Token generates token (2DH) (undefined if endpoint is an interrupt, the queue head is non-zero) transfer type, for example, µ*Frame S-mask* field in
	11b	Reserved
7-0	Status. This field is used by the Host Controller to communicate individual command execution states back to HCD. This field contains the status of the last transaction performed on this qTD. The bit encodings are:	
	Bit	Status Field Description
	7	Active. Set to one by software to enable the execution of transactions by the Host Controller.
	6	Halted. Set to one by the Host Controller during status updates to indicate that a serious error has occurred at the device/endpoint addressed by this qTD. This can be caused by babble, the error counter counting down to zero, or reception of the STALL handshake from the device during a transaction. Any time that a transaction results in the Halted bit being set to a one, the Active bit is also set to zero.
	5	Data Buffer Error. Set to a one by the Host Controller during status update to indicate that the Host Controller is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (under run). If an overrun condition occurs, the Host Controller forces a timeout condition on the USB, invalidating the transaction at the source. If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	4	Babble Detected. Set to a one by the Host Controller during status update when" babble" is detected during the transaction. In addition to setting this bit, the Host Controller also sets the *Halted* bit to a one. Because "babble" is considered a fatal error for the transfer, setting the Halted bit to a one insures that no more transactions occur because of this descriptor.
	3	Transaction Error (XactErr). Set to a one by the Host Controller during status update in the case where the host did not receive a valid response from the device (Timeout, CRC, Bad PID, etc.). If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	2	Missed Micro-Frame. This bit is ignored unless the *QH.EPS* field indicates a full- or low-speed endpoint and the queue head is in the periodic list. This bit is set when the host controller detected that a host-induced hold-off caused the host controller to miss a required complete-split transaction. If the host controller sets this bit to a one, then it remains a one for the duration of the transfer.
	1	Split Transaction State (SplitXstate). This bit is ignored by the host controller unless the *QH.EPS* field indicates a full- or low-speed endpoint. When a Full- or Low-speed device, the host controller uses this bit to track the state of the split- transaction. The functional requirements of the host controller for managing this state bit and the split transaction protocol depends on whether the endpoint is in the periodic or asynchronous schedule. The bit encodings are: Value Meaning 0b Do Start Split. This value directs the host controller to issue a Start split transaction to the endpoint. 1b Do Complete Split. This value directs the host controller to issue a Complete split transaction to the endpoint.

Table continues on the next page...

Table 44-166. TD Token (DWord 2) (continued)

Bit	Description	
	0	Ping State (P)/ERR. If the *QH.EPS* field indicates a High-speed device and the *PID_Code* indicates an OUT endpoint, then this is the state bit for the Ping protocol. The bit encodings are: Value Meaning 0b Do OUT. This value directs the host controller to issue an OUT PID to the endpoint. 1b Do Ping. This value directs the host controller to issue a PING PID to the endpoint. If the *QH.EPS* field does not indicate a High-speed device, then this field is used as an error indicator bit. It is set to a one by the host controller whenever a periodic split-transaction receives an ERR handshake.

44.6.2.5.4 qTD Buffer Page Pointer List

The last five DWords of a queue element transfer descriptor is an array of physical memory address pointers. These pointers reference the individual pages of a data buffer.

System software initializes Current Offset field to the starting offset into the current page, where current page is selected through the value in the C_Page field.

Table 44-167 describes the qTD Buffer Pointer(s) (DWords 3-7) fields.

Table 44-167. qTD Buffer Pointer(s) (DWords 3-7)

Bit	Description
31-12	Buffer Pointer List. Each element in the list is a 4 K page aligned physical memory address. The lower 12 bits in each pointer are reserved (except for the first one), as each memory pointer must reference the start of a 4 K page. The field C_Page specifies the current active pointer. When the transfer element descriptor is fetched, the starting buffer address is selected using C_Page (similar to an array index to select an array element). If a transaction spans a 4K buffer boundary, the host controller must detect the page-span boundary in the data stream, increment C_Page and advance to the next buffer pointer in the list, and conclude the transaction through the new buffer pointer.
11-0	Current Offset (Reserved). This field is reserved in all pointers except the first one (for example Page 0). The host controller should ignore all reserved bits. For the page 0 current offset interpretation, this field is the byte offset into the current page (as selected by C_Page). The host controller is not required to write this field back when the qTD is retired. Software should ensure the Reserved fields are initialized to zero.

44.6.2.6 Queue Head

Table 44-168 shows the Queue head structure layout.

Table 44-168. Queue Head Structure Layout

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Queue Head Horizontal Link Pointer																											0		Typ		T	03-00 H

Table continues on the next page...

Table 44-168. Queue Head Structure Layout (continued)

RL	C	Maximum Packet Length	H	dtc	EP	EndPt	I	Device Address	07-04H
Mult	Port Number*	Hub Addr*	µFrame C-mask*	µFrame S-mask*					0B-08H
Current qTD Pointer	0								0F-0CH
Next qTD Pointer	0	T							13-10H
Alternate Next qTD pointer	NakCnt	T							17-14H
dt	Total Bytes to Transfer	ioc	C_Page	Cerr	PID Code	Status			1B-18H
Buffer Pointer (Page 0)	Current Offset								1F-1CH
Buffer Pointer (Page 1)	Reserved	C-prog-mask*							23-20H
Buffer Pointer (Page 2)	S-bytes*	FrameTag*							27-24H
Buffer Pointer (Page 3)	-								2B-28H
Buffer Pointer (Page 4)	-								2F-2CH

Transfer Overlay Transfer Results

Static Endpoint State

These fields are used exclusively to support split transactions to USB 2.0 Hubs

(shaded)	Host Controller Read/Write	(unshaded)	Host Controller Read Only.

44.6.2.6.1 Queue Head Horizontal Link Pointer

The first DWord of a Queue Head contains a link pointer to the next data object to be processed after any required processing in this queue has been completed, as well as the control bits defined below.

This pointer may reference a queue head or one of the isochronous transfer descriptors. It must not reference a queue element transfer descriptor.

Table 44-169 describes the Queue head DWord 0 fields.

Table 44-169. Queue Head DWord 0

Bit	Description
31-5	Queue Head Horizontal Link Pointer (QHLP). This field contains the address of the next data object to be processed in the horizontal list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	QH/(s)iTD Select (Typ). This field indicates to the hardware whether the item referenced by the link pointer is an iTD, siTD or a QH. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor) 11b FSTN (frame span traversal node)
0	Terminate (T). 1=Last QH (pointer is invalid). 0=Pointer is valid. If the queue head is in the context of the periodic list, a one bit in this field indicates to the host controller that this is the end of the periodic list. This bit is ignored by the host controller when the queue head is in the Asynchronous schedule. Software must ensure that queue heads reachable by the host controller always have valid horizontal link pointers.

44.6.2.6.2 Queue Head Endpoint Capabilities/Characteristics

The second and third DWords of a Queue Head specifies static information about the endpoint. This information does not change over the lifetime of the endpoint. There are three types of information in this region:

- Endpoint Characteristics. These are the USB endpoint characteristics including addressing, maximum packet size, and endpoint speed.
- Endpoint Capabilities. These are adjustable parameters of the endpoint. They effect how the endpoint data stream is managed by the host controller.
- Split Transaction Characteristics. This data structure is used to manage full- and low-speed data streams for bulk, control, and interrupt via split transactions to USB2.0 Hub Transaction Translator. There are additional fields used for addressing the hub and scheduling the protocol transactions (for periodic).

The host controller must not modify the bits in this region.

Table 44-170 describes the Endpoint characteristics: Queue head DWord 1 fields.

Table 44-170. Endpoint Characteristics: Queue Head DWord 1

Bit	Description
31-28	Nak Count Reload (RL). This field contains a value, which is used by the host controller to reload the Nak Counter field.
27	Control Endpoint Flag (C). If the *QH.EPS* field indicates the endpoint is not a high-speed device, and the endpoint is a control endpoint, then software must set this bit to a one. Otherwise, it should always set this bit to zero.
26-16	Maximum Packet Length. This directly corresponds to the maximum packet size of the associated endpoint (*wMaxPacketSize*). The maximum value this field may contain is 0x400 (1024).
15	Head of Reclamation List Flag (H). This bit is set by System Software to mark a queue head as being the head of the reclamation list.

Table continues on the next page...

Table 44-170. Endpoint Characteristics: Queue Head DWord 1 (continued)

<table>
<tr><td>14</td><td colspan="2">Data Toggle Control (DTC). This bit specifies where the host controller should get the initial data toggle on an overlay transition.
0b Ignore DT bit from incoming qTD. Host controller preserves DT bit in the queue head.
1b Initial data toggle comes from incoming qTD DT bit. Host controller replaces DT bit in the queue head from the DT bit in the qTD.</td></tr>
<tr><td rowspan="7">13-12</td><td colspan="2">Endpoint Speed (EPS). This is the speed of the associated endpoint. Bit combinations are:</td></tr>
<tr><td>Value</td><td>Meaning</td></tr>
<tr><td>00b</td><td>Full-Speed (12 Mbits/sec)</td></tr>
<tr><td>01b</td><td>Low-Speed (1.5 Mbits/sec)</td></tr>
<tr><td>10b</td><td>High-Speed (480 Mbits/sec)</td></tr>
<tr><td>11b</td><td>Reserved</td></tr>
<tr><td colspan="2">This field must not be modified by the host controller.</td></tr>
<tr><td>11-8</td><td colspan="2">Endpoint Number (Endpt). This 4-bit field selects the particular endpoint number on the device serving as the data source or sink.</td></tr>
<tr><td>7</td><td colspan="2">Inactivate on Next Transaction (I). This bit is used by system software to request that the host controller set the Active bit to zero. See Rebalancing the Periodic Schedule , for full operational details. This field is only valid when the queue head is in the Periodic Schedule and the EPS field indicates a Full or Low-speed endpoint. Setting this bit to one when the queue head is in the Asynchronous Schedule or the EPS field indicates a high-speed device yields undefined results.</td></tr>
<tr><td>6-0</td><td colspan="2">Device Address. This field selects the specific device serving as the data source or sink.</td></tr>
</table>

Table 44-171 describes the Endpoint capabilities: Queue head DWord 2 field descriptions.

Table 44-171. Endpoint Capabilities: Queue Head DWord 2

Bit	Description
31-30	High-Bandwidth Pipe Multiplier (Mult). This field is a multiplier used to key the host controller as the number of successive packets the host controller may submit to the endpoint in the current execution. The host controller makes the simplifying assumption that software properly initializes this field (regardless of location of queue head in the schedules or other run time parameters). The valid values are: Value Meaning 00b Reserved. A zero in this field yields undefined results. 01b One transaction to be issued for this endpoint per micro-frame 10b Two transactions to be issued for this endpoint per micro-frame 11b Three transactions to be issued for this endpoint per micro-frame
29-23	Port Number. This field is ignored by the host controller unless the *EPS* field indicates a full- or low-speed device. The value is the port number identifier on the USB 2.0 Hub (for hub at device address *Hub Addr* below), below which the full- or low-speed device associated with this endpoint is attached. This information is used in the split-transaction protocol.
22-16	Hub Addr. This field is ignored by the host controller unless the *EPS* field indicates a full-or low-speed device. The value is the USB device address of the USB 2.0 Hub below which the full- or low-speed device associated with this endpoint is attached. This field is used in the split-transaction protocol.

Table continues on the next page...

Table 44-171. Endpoint Capabilities: Queue Head DWord 2 (continued)

15-8	Split Completion Mask (μFrame C-Mask). This field is ignored by the host controller unless the *EPS* field indicates this device is a low- or full-speed device and this queue head is in the periodic list. This field (along with the *Active* and *SplitX-state* fields) is used to determine during which micro-frames the host controller should execute a complete-split transaction. When the criteria for using this field are met, a zero value in this field has undefined behavior. This field is used by the host controller to match against the three low-order bits of the FRINDEX register. If the FRINDEX register bits decode to a position where the μ*Frame C- Mask* field is a one, then this queue head is a candidate for transaction execution. There may be more than one bit in this mask set.
7-0	Interrupt Schedule Mask (μFrame S-mask). This field is used for all endpoint speeds. Software should set this field to a zero when the queue head is on the asynchronous schedule. A non-zero value in this field indicates an interrupt endpoint. The host controller uses the value of the three low-order bits of the FRINDEX register as an index into a bit position in this bit vector. If the μ*Frame S-mask* field has a one at the indexed bit position then this queue head is a candidate for transaction execution. If the *EPS* field indicates the endpoint is a high-speed endpoint, then the transaction executed is determined by the *PID_Code* field contained in the execution area. This field is also used to support split transaction types: Interrupt (IN/OUT). This condition is true when this field is non-zero and the *EPS* field indicates this is either a full- or low-speed device. A zero value in this field, in combination with existing in the periodic frame list has undefined results.

44.6.2.6.3 Transfer Overlay-Queue Head

The nine DWords in this area represent a transaction working space for the host controller. The general operational model is that the host controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, then it follows the Queue Head Horizontal Link Pointer to the next queue head. The host controller will never follow the Next Transfer Queue Element or Alternate Queue Element pointers unless it is actively attempting to advance the queue. For the duration of the transfer, the host controller keeps the incremental status of the transfer in the overlay area. When the transfer is complete, the results are written back to the original queue element.

The DWord3 of a Queue Head contains a pointer to the source qTD currently associated with the overlay. The host controller uses this pointer to write back the overlay area into the source qTD after the transfer is complete.

Table 44-172 describes the current qTD link pointer field descriptions.

Table 44-172. Current qTD Link Pointer

Bit	Description
31-5	Current Element Transaction Descriptor Link Pointer. This field contains the address Of the current transaction being processed in this queue and corresponds to memory address signals [31:5], respectively.
4-0	Reserved (R). These bits are ignored by the host controller when using the value as an address to write data. The actual value may vary depending on the usage.

The DWords 4-11 of a queue head are the transaction overlay area. This area has the same base structure as a Queue Element Transfer Descriptor. The queue head utilizes the reserved fields of the page pointers to implement tracking the state of split transactions.

This area is characterized as an overlay because when the queue is advanced to the next queue element, the source queue element is merged onto this area. This area serves an execution cache for the transfer.

Table 44-173 describes the Host-controller rules for bits in overlay.

Table 44-173. Host-Controller Rules for Bits in Overlay (DWords 5, 6, 8 and 9)

DWord	Bit	Description
5	4-1	Nak Counter (NakCnt)µRW. This field is a counter the host controller decrements whenever a transaction for the endpoint associated with this queue head results in a Nak or Nyet response. This counter is reloaded from *RL* before a transaction is executed during the first pass of the reclamation list (relative to an Asynchronous List Restart condition). It is also loaded from RL during an overlay.
6	31	Data Toggle. The *Data Toggle Control* controls whether the host controller preserves this bit when an overlay operation is performed.
6	15	Interrupt On Complete (IOC). The IOC control bit is always inherited from the source qTD when the overlay operation is performed.
6	11-10	Error Counter (C_ERR). This 2-bit field is copied from the qTD during the overlay and written back during queue advancement.
6	0	Ping State (P)/ERR. If the *EPS* field indicates a high-speed endpoint, then this field should be preserved during the overlay operation.
8	7-0	Split-transaction Complete-split Progress (C-prog-mask). This field is initialized to zero during any overlay. This field is used to track the progress of an interrupt split-transaction.
9	4-0	Split-transaction Frame Tag (Frame Tag). This field is initialized to zero during any overlay. This field is used to track the progress of an interrupt split-transaction.
9	11-5	S-bytes. Software must ensure that the *S-bytes* field in a *qTD* is zero before activating the *qTD*. This field is used to keep track of the number of bytes sent or received during an IN or OUT split transaction.

44.6.2.7 Periodic Frame Span Traversal Node (FSTN)

This data structure is to be used only for managing Full- and Low-speed transactions that span a Host-frame boundary. See Host Controller Operational Model for FSTNs for full operational details. Software must not use an FSTN in the Asynchronous Schedule. An FSTN in the Asynchronous schedule results in undefined behavior. Software must not use the FSTN feature with a host controller whose USB_HCIVERSION register indicates a revision implementation below 0096h. FSTNs are not defined for implementations before 0.96 and their use yields undefined results.

Table continues on the next page...

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	Normal Path Link Pointer																											0		Typ		T	03-00H
	Back Path Link Pointer																											0		Typ		T	07-04H

Table 44-174. Frame Span Traversal Node Structure Layout

	Host Controller Read/Write		Host Controller Read Only.

44.6.2.7.1 FSTN Normal Path Pointer

The first DWord of an FSTN contains a link pointer to the next schedule object. This object can be of any valid periodic schedule data type.

Table 44-175 describes the FSTN normal path pointer fields.

Table 44-175. FSTN Normal Path Pointer Field Descriptions

Bit	Description
31-5	Normal Path Link Pointer (NPLP). This field contains the address of the next data object to be processed in the periodic list and corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	QH/(s)iTD/FSTN Select (Typ). This field indicates to the Host Controller whether the item referenced is a iTD/siTD, a QH or an FSTN. This allows the Host Controller to perform the proper type of processing on the item after it is fetched. Value encodings are: Value Meaning 00b iTD (isochronous transfer descriptor) 01b QH (queue head) 10b siTD (split transaction isochronous transfer descriptor) 11b FSTN (Frame Span Traversal Node)
0	Terminate (T). 1=Link Pointer field is not valid. 0=Link Pointer is valid.

44.6.2.7.2 FSTN Back Path Link Pointer

The second DWord of an FTSN node contains a link pointer to a queue head. If the T-bit in this pointer is zero, then this FSTN is a Save-Place indicator. Its Typ field must be set by software to indicate the target data structure is a queue head. If the T-bit in this pointer is set to one, then this FSTN is the Restore indicator. When the T-bit is one, the host controller ignores the Typ field.

Table 44-176 describes the FSTN back path link pointer fields.

Table 44-176. FSTN Back Path Link Pointer Field Descriptions

Bit	Description
31-5	Back Path Link Pointer (BPLP). This field contains the address of a Queue Head. This field corresponds to memory address signals [31:5], respectively.
4-3	Reserved
2-1	Typ. Software must ensure this field is set to indicate the target data structure is a Queue Head. Any other value in this field yields undefined results.
0	Terminate (T). 1=Link Pointer field is not valid (that is the host controller must not use bits [31:5] as a valid memory address). This value also indicates that this FSTN is a Restore indicator. 0=Link Pointer is valid (that is the host controller may use bits [31:5] (in combination with the CTRLDSSEGMENT register if applicable) as a valid memory address). This value also indicates that this FSTN is a Save-Place indicator.

44.6.3 Host Operational Model

The general operational model is for the enhanced interface host controller hardware and enhanced interface host controller driver (generally referred to as system software). Each significant operational feature of the EHCI host controller is discussed in a separate section. Each section presents the operational model requirements for the host controller hardware. Where appropriate, recommended system software operational models for features are also presented.

44.6.3.1 Host Controller Initialization

When the system boots, the host controller is enumerated, assigned a base address for the register space and BIOS sets the USB_FLADJ register to a system-specific value. After initial power-on or HCReset (hardware or through HCReset bit in the USB_USBCMD register), all of the operational registers are at their default values. After a hardware reset, only the operational registers not contained in the Auxiliary power well are at their default values.

Table 44-177 describes the default values of operational registers.

.

Table 44-177. Default Values of Operational Register Space

Operational Register	Default Value (after Reset)
USB_USBCMD	00080000h (00080B00h, if *Asynchronous Schedule Park Capability is one*)
USB_USBSTS	00001000h
USB_USBINTR	00000000h

Table continues on the next page...

Table 44-177. Default Values of Operational Register Space (continued)

USB_FRINDEX	00000000h
USB_CTRLDSSEGMENT	00000000h
USB_PERIODICLISTBASE	Undefined
USB_ASYNCLISTADDR	Undefined
USB_CONFIGFLAG	00000000h
USB_PORTSC1	00002000h (w/*PPC* set to one); 00003000h (w/*PPC* set to zero)

To initialize the host controller, software should perform the following steps:

- Reset the controller by setting the USBCMD:RST bit and wait for it to clear. Configure the controller for HOST operation by setting USBMODE.CMD to 3.
- Program the USB_CTRLDSSEGMENT register with 4-Gbyte segment where all of the interface data structures are allocated.
- Write the appropriate value to the USB_USBINTR register to enable the appropriate interrupts.
- Write the base address of the Periodic Frame List to the USB_PERIODICLIST BASE register. If no work items are in the periodic schedule, all elements of the Periodic Frame List should have their T-Bits set to one.
- Write the USB_USBCMD register to set the desired interrupt threshold, frame list size (if applicable) and turn the host controller ON through setting the Run/Stop bit.

At this point, the host controller is up and running and the port registers begin reporting device connects, and so on. System software can enumerate a port through the reset process (where the port is in the enabled state). At this point, the port is active with SOFs occurring down the enabled port enabled High-speed ports, but the schedules have not enabled. The EHCI Host controller do not transmit SOFs to enabled Full- or Low-speed ports. To communicate with devices through the asynchronous schedule, system software must write the USB_ASYNDLISTADDR register with the address of a control or bulk queue head. Software must then enable the asynchronous schedule by writing one to the Asynchronous Schedule Enable bit in the USB_USBCMD register. To communicate with devices through the periodic schedule, system software must enable the periodic schedule by writing one to the Periodic Schedule Enable bit in the USB_USBCMD register.

NOTE

The schedules can be turned on before the first port is reset (and enabled).

When the USB_USBCMD register is written, system software must ensure the appropriate bits are preserved, depending on the intended operation.

44.6.3.2 Port Routing and Control

A USB 2.0 Host controller is comprised of one high-speed host controller, which implements the EHCI programming interface and 0 to N USB 1.1 companion host controllers. Companion host controllers (cHCs) may be implementations of either Universal or Open host controller specifications. This configuration is used to deliver the required full USB 2.0-defined port capability; for example, Low-, Full-, and High-speed capability for every port.

NOTE

The USB controller on this part does not require nor support companion controllers to support Full Speed/Low Speed devices. Therefore, no port routing is present in the controller. Please refer to Embedded Transaction Translator Function for detail!

Figure 44-142 illustrates a simple block diagram of the port routing logic and its relationship to the high-speed and companion host controllers within a USB 2.0 host controller.

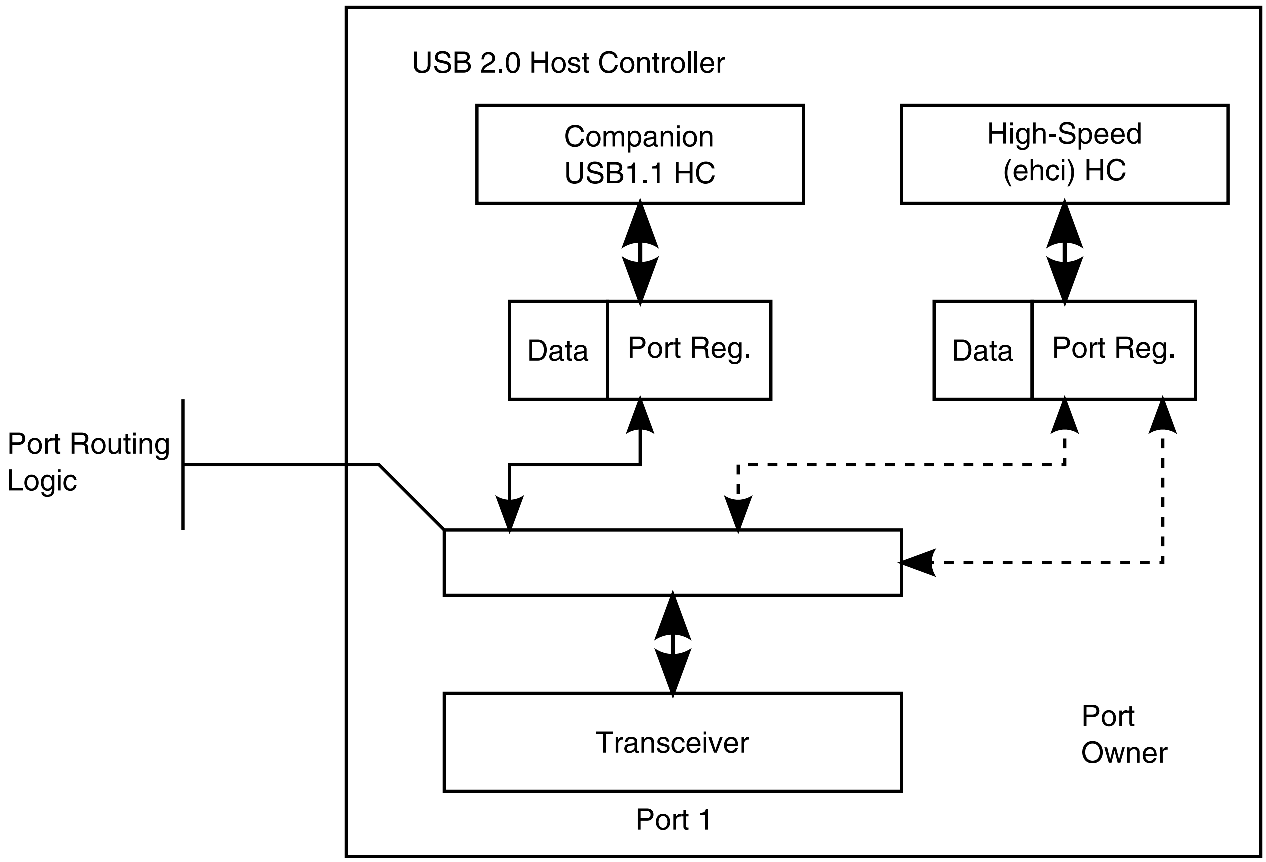


Figure 44-142. Example USB 2.0 Host Controller Port Routing Block Diagram

There exists one transceiver per physical port and each host controller block has its own port status and control registers. The EHCI controller has port status and control registers for every port. Each companion host controller has only the port control and status registers it is required to operate. Either the EHCI host controller or one companion host controller controls each transceiver. Routing logic lies between the transceiver, the port status and control registers.[4]

The port routing logic is controlled from signals originating in the EHCI host controller. The EHCI host controller has a global routing policy control field and per-port ownership control fields. The Configured Flag (CF) bit is the global routing policy control. At power-on or reset, the default routing policy is to the companion controllers (if they exist). If the system does not include a driver for the EHCI host controller and the host controller includes Companion Controllers, then the ports still work in Full- and Low-speed mode (assuming the system includes a driver for the companion controllers). In general, when the EHCI owns the ports, the companion host controllers' port registers do not see a connect indication from the transceiver. Similarly, when a companion host controller owns a port, the EHCI controller's port registers do not see a connect indication from the transceiver. The details on the rules for the port routing logic are described in the following sections. The USB 2.0 host controller must be implemented as a multi-function PCI device if the implementation includes companion controllers. The companion host controllers' function numbers must be less than the EHCI host controller function number. The EHCI host controller must be a larger function number with respect to the companion host controllers associated with this EHCI host controller. If a PCI device implementation contains only an EHCI controller (that is no companion controllers or other PCI functions), then the EHCI host controller must be function zero, in accordance with the PCI Specification. The N_CC field in the Structural Parameter register (HCSPARAMS) indicates whether the controller implementation includes companion host controllers. When N_CC has a non-zero value there exists companion host controllers. If N_CC has a value of zero, then the host controller implementation does not include companion host controllers. If the host controller root ports are exposed to attachment of full- or low-speed devices, the ports always fails the high-speed chirp during reset and the ports are not enabled. System software can notify the user of the illegal condition. This type of implementation requires a USB 2.0 hub be connected to a root port to provide full and low-speed device connectivity.

System software uses information in the host controller capability registers to determine how the ports are routed to the companion host controllers. See Host Controller Structural Parameters (USB_HCSPARAMS).

4. The routing logic should not be implemented in the 480 MHz clock domain of the transceiver.

44.6.3.2.1 Port Routing Control through EHCI Configured (CF) Bit

Each port in the USB 2.0 host controller are routed either to a single companion host controller or to the EHCI host controller. The port routing logic is controlled by two mechanisms in the EHCI HC: a host controller global flag and per-port control. The Configured Flag (CF) bit, is used to globally set the policy of the routing logic. Each port register has a Port Owner control bit which allows the EHCI Driver to explicitly control the routing of individual ports. Whenever the CF bit transitions from zero to one (this transition is only available under program control) the port routing unconditionally routes all of the port registers to the EHCI HC (all Port Owner bits go to zero). While the CF-bit is one, the EHCI Driver controls individual ports' routing through the Port Owner control bit. Likewise, whenever the CF bit transitions from one to zero (as a result of Aux power application, HCRESET, or software writing zero to CF-bit), the port routing unconditionally routes all of the port registers to the appropriate companion HC. The default value for the EHCI HC's CF bit (after Aux power application or HCRESET) is zero.

The *view* of the port depends on the current owner. A Universal or Open companion host controller will see port register bits consistent with the appropriate specification. Port bit definitions that are required for EHCI host controllers are not visible to companion host controllers.

Table 44-178 summarizes the default routing for all the ports, based on the value of the EHCI HC's CF bit.

Table 44-178. Default Port Routing Depending on EHCI HC CF Bit

HS CF Bit	Default Port Ownership	Explanation
0B	Companion HCs	The companion host controllers own the ports and only Full- and Low-speed devices are supported in the system. The exact port assignments are implementation dependent. The ports behave only as Full- and Low-speed ports in this configuration
1B	EHCI HC	The EHCI host controller has default ownership over all of the ports. The routing logic inhibits device connect events from reaching the companion HCs' port status and control registers when the port owner is the EHCI HC.The EHCI HC has access to the additional port status and control bits defined in this specification (see Port Status & Control (USB_PORTSC1)). The EHCI HC can temporarily release control of the port to a companion HC by setting the *PortOwner* bit in the PORTSC1 register to one.

44.6.3.2.2 Port Routing Control through PortOwner and Disconnect Event

Manipulating the port routing through the CF-bit is an extreme process and not intended to be used during normal operation. The normal mode of port ownership transferal is on the granularity of individual ports using the Port Owner bit in the EHCI HC's

USB_PORTSC1 register (for hand-offs from EHCI to companion host controllers). Individual port ownership is returned to the EHCI controller when the port registers a device disconnect. When the disconnect is detected, the port routing logic immediately returns the port ownership to the EHCI controller. The companion host controller port register detects the device disconnect and operates normally.

Under normal operating conditions (assuming all HC drivers loaded and operational and the EHCI *CF-bit* is set to one), the typical port enumeration sequence proceeds as illustrated below:

- Initial condition is that EHCI is port owner. A device is connected causing the port to detect a connect, set the port connect change bit and issue a port-change interrupt (if enabled).
- EHCI Driver identifies the port with the new connect change bit asserted and sends a change report to the hub driver. Hub driver issues a GetPortStatus() request and identifies the connect change. It then issues a request to clear the connect change, followed by a request to reset and enable the port.
- When the EHCI Driver receives the request to reset and enable the port, it first checks the value reported by the LineStatus bits in the USB_PORTSC1 register. If they indicate the attached device is a full-speed device (for example, D+ is asserted), then the EHCI Driver sets the PortReset control bit to one (and sets the PortEnable bit to zero) which begins the reset-process. Software times the duration of the reset, then terminates reset signaling by writing zero to the port reset bit. The reset process is actually complete when software reads zero in the PortReset bit. The EHCI Driver checks the PortOwner bit in the USB_PORTSC1 register. If set to one, the connected device is a high-speed device and EHCI Driver (root hub emulator) issues a change report to the hub driver and the hub driver continues to enumerate the attached device.
- At the time the EHCI Driver receives the port reset and enable request the LineStatus bits might indicate a low-speed device. Additionally, when the port reset process is complete, the PortEnable field may indicate that a full-speed device is attached. In either case the EHCI driver sets the PortOwner bit in the USB_PORTSC1 register to one to release port ownership to a companion host controller.
- When the EHCI Driver sets PortOwner bit to one, the port routing logic makes the connection state of the transceiver available to the companion host controller port register and removes the connection state from the EHCI HC port. The EHCI USB_PORTSC1 register observes and reports a disconnect event through the disconnect change bit. The EHCI Driver detects the connection status change (either by polling or by port change interrupt) and then sends a change report to the hub driver. When the hub driver requests that port-state, the EHCI Driver responds with a reset complete change set to one, a connect change set to one and a connect status set to zero. This information is derived directly from the EHCI port register. This allows

the hub driver to assume the device was disconnected during reset. It acknowledges the change bits and wait for the next change event. While the EHCI controller does not own the port, it simply remains in a state where the port reports no device connected. The device-connect evaluation circuitry of the companion HC activates and detects the device, the companion Driver detects the connection and enumerates the port.

When a port is routed to a companion HC, it remains under the control of the companion HC until the device is disconnected from the root port (ignoring for now the scenario where EHCI's CF-bit transitions from1b to 0b). When a disconnect occurs, the disconnect event is detected by both the companion HC port control and the EHCI port ownership control. On the event, the port ownership is returned immediately to the EHCI controller. The companion HC stack detects the disconnect and acknowledges as it would in an ordinary standalone implementation. Subsequent connects is detected by the EHCI port register and the process repeats.

44.6.3.2.3 Example Port Routing State Machine

Figure 44-143 illustrates an example of how the port ownership should be managed. The following sections describe the entry conditions to each state.

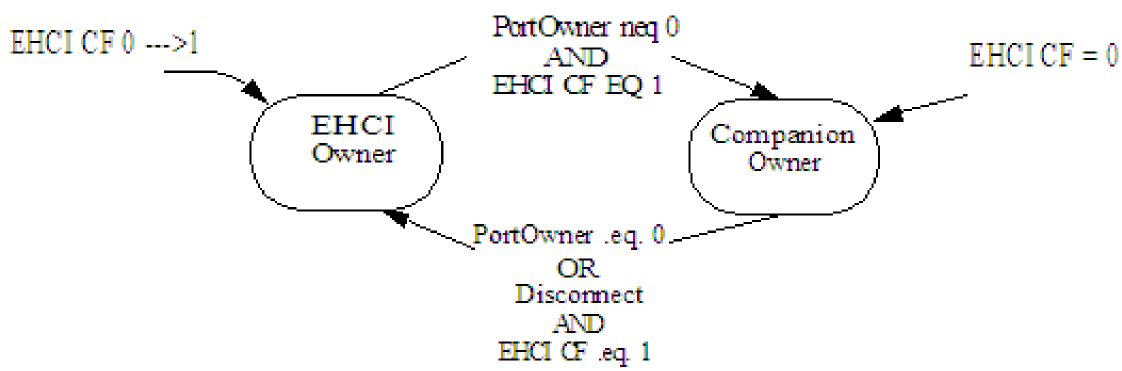


Figure 44-143. Port Owner Handoff State Machine

44.6.3.2.3.1 EHCI HC Owner

Entry to this state occurs when one of the following events occur:

- When the EHCI HC's Configure Flag (CF) bit in the USB_CONFIGFLAG register transitions from zero to one. This signals the fact that the system has a host controller driver for the EHCI HC and that all ports in the USB 2.0 host controller must default route to the EHCI controller.

- When the port is owned by a companion HC and the device is disconnected from the port. The EHCI port routing control logic is notified of the disconnect, and returns port routing to the EHCI controller. The connection state of the companion HC goes immediately to the disconnected state (with appropriate side effect to connect change, enable and enable change). The companion HC driver acknowledges the disconnect by setting the connect status change bit to zero. This allows the companion HC's driver to interact with the port completely through the disconnect process.
- When system software writes zero to the PortOwner bit in the USB_PORTSC1 register. This allows software to take ownership of a port from a companion host controller. When this occurs, the routing logic to the companion HC effectively signals a disconnect to the companion HC's port status and control register.

44.6.3.2.3.2 Companion HC Owner

Entry to this state occurs whenever one of the following events occurs:

- When the PortOwner field transitions from zero to one.
- When the HS-mode HC's Configure Flag (CF) is equal to zero.

On entry to this state, the routing logic allows the companion HC port register to detect a device connect. Normal port enumeration proceeds.

44.6.3.2.4 Port Power

The Port Power Control (PPC) bit in the USB_HCSPARAMS register indicates whether the USB 2.0 host controller has port power control (see Host Controller Structural Parameters (USB_HCSPARAMS)). When this bit is zero, then the host controller does not support software control of port power switches. When in this configuration, the port power is always available and the companion host controllers must implement functionality consistent with port power always on. When the *PPC* bit is one, then the host controller implementation includes port power switches. Each available switch has an output enable, which is referred to in this discussion as PortPowerOutputEnable (PPE). PPE is controlled based on the state of the combination bits PPC bit, EHCI Configured (CF)-bit and individual Port Power (PP) bits.

Table 44-179 describes the summary behavioral model.

Table 44-179. Port Power Enable Control Rules

CF	CHC[1] (PP)	EHC[2] (PP)	Owner	PPE[3]	Description

Table continues on the next page...

Table 44-179. Port Power Enable Control Rules (continued)

0	0	X	CHC	0	When the EHCI controller is not configured, the port is owned by the companion host controller. When the companion HC's port power select is off, then the port power is off.
0	1	X	CHC	1	Similar to previous entry. When the companion HC's port power select is on, then the port power is on.
1	0	0	CHC	0	Port owner has port power turned off, the power to port is off.
1	0	0	EHC	0	Port owner has port power turned off, the power to port is off.
1	0	1	EHC	1	Port owner has port power on, so power to port is on.
1	0	1	CHC	1	If either HC has port power turned on, the power to the port is on.
1	1	0	EHC	1	If either HC has port power turned on, the power to the port is on.
1	1	0	CHC	1	Port owner has port power on, so power to port is on.
1	1	1	CHC	1	Port owner has port power on, so power to port is on.
1	1	1	EHC	1	Port owner has port power on, so power to port is on.

1. CHC (Companion Host Controller).
2. EHC (EHCI Host Controller).
3. PPE (Port Power Enable). This bit actually turns on the port power switch (if one exists).

44.6.3.2.5 Port Reporting Over-Current

Host controllers are by definition power providers on USB. Whether the ports are considered high- or low-powered is a platform implementation issue. Each EHCI USB_PORTSC1 register has an over-current status and over-current change bit. The functionality of these bits are specified in the USB Specification Revision 2.0.

The over current detection and limiting logic usually resides outside the host controller logic. This logic may be associated with one or more ports. When this logic detects an over-current condition it is made available to both the companion and EHCI ports. The effect of an over-current status on a companion host controller port is beyond the scope of this document.

The over-current condition effects the following bits in the USB_PORTSC1 register on the EHCI port:

- Over-current Active bits are set to one. When the over-current condition goes away, the Over-current Active bit transitions from one to zero.

- Over-current Change bits are set to one. On every transition of the Over-current Active bit the host controller sets the Over-current Change bit to one. Software sets the Over-current Change bit to zero by writing one to this bit.
- Port Enabled/Disabled bit is set to zero. When this change bit gets set to one, then the Port Change Detect bit in the USB_USBSTS register is set to one.
- Port Power (PP) bits may optionally be set to zero. There is no requirement in USB that a power provider shut off power in an over current condition. It is sufficient to limit the current and leave power applied. When the Over-current Change bit transitions from zero to one, the host controller also sets the Port Change Detect bit in the USB_USBSTS register to one. In addition, if the Port Change Interrupt Enable bit in the USB_USBINTR register is one, then the host controller issues an interrupt to the system. Refer to Table 44-180 for summary behavior for over-current detection when the host controller is halted (suspended from a device component point of view).

44.6.3.3 Suspend/Resume-Host Operational Model

The EHCI host controller provides an equivalent suspend and resume model as that defined for individual ports in a USB 2.0 Hub. Control mechanisms are provided to allow system software to suspend and resume individual ports. The mechanisms allow the individual ports to be resumed completely through software initiation. Other control mechanisms are provided to parameterize the host controller's response (or sensitivity) to external resume events. In this discussion, host-initiated, or software initiated resumes are called Resume Events/Actions. Bus-initiated resume events are called wake-up events. The classes of wake-up events are:

- Remote-wake-up enabled device asserts resume signaling. In similar kind to USB 2.0 Hubs, EHCI controllers must always respond to explicit device resume signaling and wake-up the system (if necessary).
- Port connect and disconnect and over-current events. Sensitivity to these events can be turned on or off by using the per-port control bits in the USB_PORTSC1 registers.

Selective suspend is a feature supported by every USB_PORTSC1 register. It is used to place specific ports into a suspend mode. This feature is used as a functional component for implementing the appropriate power management policy implemented in a particular operating system. When system software intends to suspend the entire bus, it should selectively suspend all enabled ports, then shut off the host controller by setting the Run/Stop bit in the USB_USBCMD register to zero. The EHCI sub-block can then be placed into a lower device state through the PCI power management interface (see Appendix A, Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 0.95, November 2000, Intel Corporation. http://www.intel.com).

When a wake event occurs, the system resumes operation and system software eventually set the Run/Stop bit to one and resume the suspended ports. Software must not set the Run/Stop bit to one until it is confirmed that the clock to the host controller is stable. This is usually confirmed in a system implementation in that all of the clocks in the system are stable before the ARM platform is restarted. So, by definition, if software is running, clocks in the system are stable and the Run/Stop bit in the USB_USBCMD register can be set to one. Minimum system software delays are also defined in the PCI Power Management Specification. Refer to PCI Power Management Specification for more information.

44.6.3.3.1 Port Suspend/Resume

System software places individual ports into suspend mode by writing one into the appropriate USB_PORTSC1 Suspend bit. Software must only set the Suspend bit when the port is in the enabled state (Port Enabled bit is one) and the EHCI is the port owner (PortOwner bit is zero).

The host controller may evaluate the Suspend bit immediately or wait until a micro-frame or frame boundary occurs. If evaluated immediately, the port is not suspended until the current transaction (if one is executing) completes. Therefore, there may be several micro-frames of activity on the port until the host controller evaluates the Suspend bit. The host controller must evaluate the Suspend bit at least every frame boundary.

System software can initiate a resume on a selectively suspended port by writing one to the Force Port Resume bit. Software should not attempt to resume a port unless the port reports that it is in the suspended state (see Port Status & Control (USB_PORTSC1)). If system software sets Force Port Resume bit to one when the port is not in the suspended state, the resulting behavior is undefined. In order to assure proper USB device operation, software must wait for at least 10 ms after a port indicates that it is suspended (Suspend bit is one) before initiating a port resume through the Force Port Resume bit. When Force Port Resume bit is one, the host controller sends resume signaling down the port. System software times the duration of the resume (nominally 20 ms) then sets the Force Port Resume bit to zero. When the host controller receives the write to transition Force Port Resume to zero, it completes the resume sequence as defined in the USB specification, and sets both the Force Port Resume and Suspend bits to zero. Software-initiated port resumes do not affect the Port Change Detect bit in the USB_USBSTS register nor do they cause an interrupt if the Port Change Interrupt Enable bit in the USB_USBINTR register is one. An external USB event may also initiate a resume. The wake events are defined above. When a wake event occurs on a suspended port, the resume signaling is detected by the port and the resume is reflected downstream within 100 μsec. The port's

Force Port Resume bit is set to one and the Port Change Detect bit in the USB_USBSTS register is set to one. If the Port Change Interrupt Enable bit in the USB_USBINTR register is one the host controller issues a hardware interrupt.

System software observes the resume event on the port, delays a port resume time (nominally 20 ms), then terminates the resume sequence by writing zero to the Force Port Resume bit in the port. The host controller receives the write of zero to Force Port Resume, terminates the resume sequence and sets Force Port Resume and Suspend port bits to zero. Software can determine that the port is enabled (not suspended) by sampling the USB_PORTSC1 register and observing that the Suspend and Force Port Resume bits are zero. Software must ensure that the host controller is running (that is HCHalted bit in the USB_USBSTS register is zero), before terminating a resume by writing zero to a port's Force Port Resume bit. If HCHalted is one when Force Port Resume is set to zero, then SOFs do not occur down the enabled port and the device returns to suspend mode in a maximum of 10 msec.

Table 44-180 summarizes the wake-up events. Whenever a resume event is detected, the Port Change Detect bit in the USB_USBSTS register is set to one. If the Port Change Interrupt Enable bit is one in the USB_USBINTR register, the host controller generates an interrupt on the resume event. Software acknowledges the resume event interrupt by clearing the Port Change Detect status bit in the USB_USBSTS register.

Table 44-180. Behavior During Wake-up Events

Port Status and Signaling Type	Signaled Port Response	Device State	
		D0	Not D0
Port disabled, resume K-State received	No Effect	N/A	N/A
Port suspended, resume K-State received	Resume reflected downstream on signaled port. Force Port Resume status bit in USB_PORTSC1 register is set to one. Port Change Detect bit in USB_USBSTS register set to one.	[1], [2]	[2]
Port is enabled, disabled or suspended, and the port's WKDSCNNT_E bit is one. A disconnect is detected.	Depending in the initial port state, theUSB_ PORTSC1 Connected Enable status bits are set to zero, and the Connect Change status bit is set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [2]	[2]
Port is enabled, disabled or suspended, and the port's WKDSCNNT_E bit is zero. A disconnect is detected.	Depending on the initial port state, the USB_PORTSC1 Connect and Enable status bits are set to zero, and the Connect Change status bit is set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]
Port is not connected and the port's WKCNNT_E bit is one. A connect is detected.	USB_PORTSC1 Connect Status and Connect Status Change bits are set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [2]	[2]
Port is not connected and the port's WKCNNT_E bit is zero. A connect is detected.	USB_PORTSC1 Connect Status and Connect Status Change bits are set to one. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]

Table continues on the next page...

Table 44-180. Behavior During Wake-up Events (continued)

Port is connected and the port's WKOC_E bit is one. An over-current condition occurs.	USB_PORTSC1 Over-current Active, Over-current Change bits are set to one. If Port Enable/Disable bit is one, it is set to zero. Port Change Detect bit in the USB_USBSTS register is set to one	[1], [2]	[2]
Port is connected and the port's WKOC_E bit is zero. An over-current condition occurs.	USB_PORTSC1 Over-current Active, Over-current Change bits are set to one. If Port Enable/Disable bit is one, it is set to zero. Port Change Detect bit in the USB_USBSTS register is set to one.	[1], [3]	[3]

[1] Hardware interrupt issued if Port Change Interrupt Enable bit in the USB_USBINTR register is one.

[2] PME# asserted if enabled (Note: PME Status must always be set to one).

[3] PME# not asserted.

44.6.3.4 Schedule Traversal Rules

The host controller executes transactions for devices using a simple, shared-memory schedule. The schedule is comprised of a few data structures, organized into two distinct lists. The data structures are designed to provide the maximum flexibility required by USB, minimize memory traffic and hardware / software complexity.

System software maintains two schedules for the host controller: a periodic schedule and an asynchronous schedule. The root of the periodic schedule is the USB_PERIODICLISTBASE register (see Frame List Base Address (USB_PERIODICLISTBASE))/Device Address (USB_DEVICEADDR) The USB_PERIODICLISTBASE register is the physical memory base address of the periodic frame list. The periodic frame list is an array of physical memory pointers. The objects referenced from the frame list must be valid schedule data structures as defined in Host data structures. In each micro-frame, if the periodic schedule is enabled (see Periodic Scheduling Threshold) then the host controller must execute from the periodic schedule before executing from the asynchronous schedule. It only executes from the asynchronous schedule after it encounters the end of the periodic schedule. The host controller traverses the periodic schedule by constructing an array offset reference from the USB_PERIODICLISTBASE and the USB_FRINDEX registers (see Figure 44-144). It fetches the element and begins traversing the graph of linked schedule data structures.

The end of the periodic schedule is identified by a next link pointer of a schedule data structure having its T-bit set to one. When the host controller encounters a T-Bit set to one during a horizontal traversal of the periodic list, it interprets this as an End-Of-Periodic-List mark. This causes the host controller to cease working on the periodic

schedule and transitions immediately to traversing the asynchronous schedule. After the transition, the host controller executes from the asynchronous schedule until the end of the micro-frame.

Figure 44-144 illustrates the derivation of pointer into frame list array.

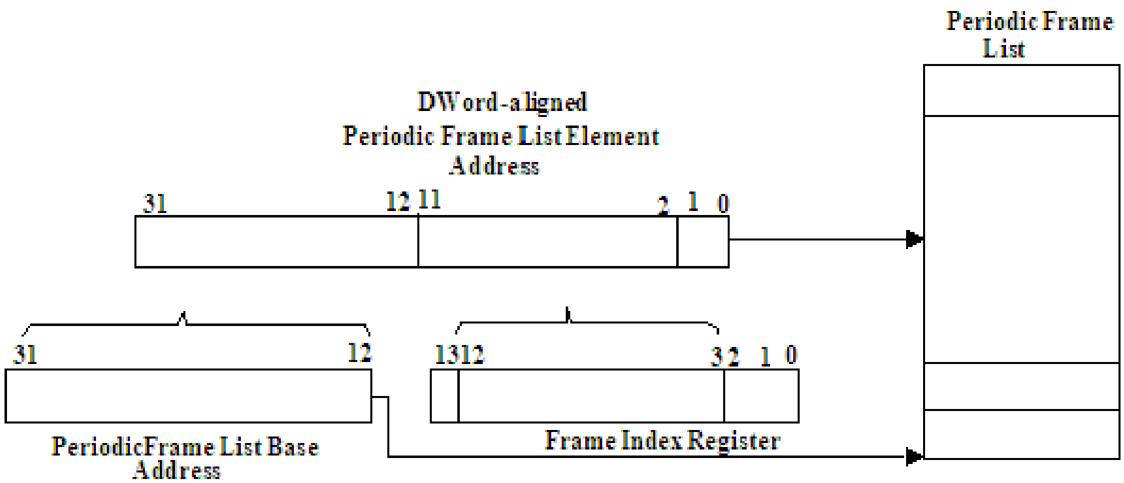


Figure 44-144. Derivation of Pointer into Frame List Array

When the host controller determines that it is the time to execute from the asynchronous list, it uses the operational register USB_ASYNCLISTADDR to access the asynchronous schedule, see Figure 44-145.

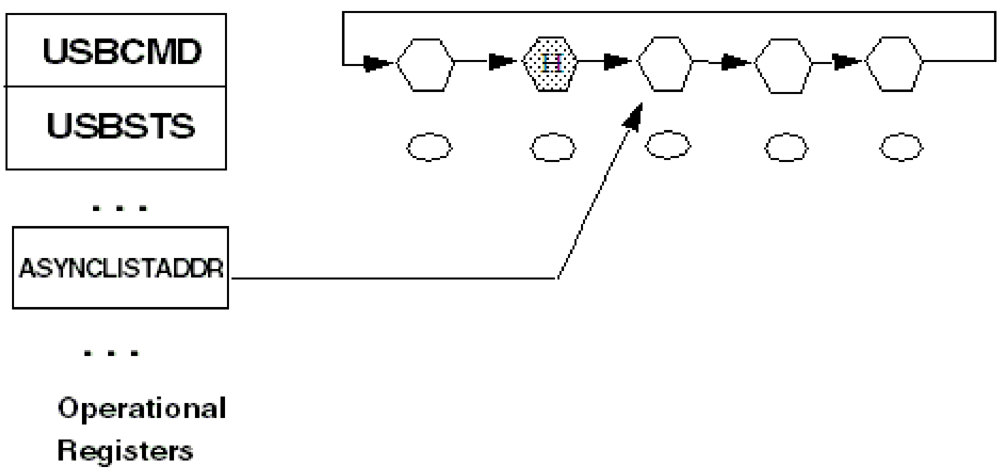


Figure 44-145. General Format of Asynchronous Schedule List

The USB_ASYNCLISTADDR register contains a physical memory pointer to the next queue head. When the host controller makes a transition to executing the asynchronous schedule, it begins by reading the queue head referenced by the

USB_ASYNCLISTADDR register. Software must set queue head horizontal pointer T-bits to zero for queue heads in the asynchronous schedule. See Asynchronous Schedule for complete operational details.

44.6.3.4.1 Example - Preserving Micro-Frame Integrity

One of the requirements of a USB host controller is to maintain Frame Integrity. This means that the HC must preserve the micro-frame boundaries. For example, SOF packets must be generated on time (within the specified allowable jitter), and High-speed EOF1,2 thresholds must be enforced. The end of micro-frame timing points EOF1 and EOF2 are clearly defined in the USB Specification Revision 2.0. One implication of this responsibility is that the HC must ensure that it does not start transactions that do not complete before the end of the micro-frame. More precisely, no transactions should be started by the host controller, which do not complete in their entirety before the EOF1 point. In order to enforce this rule, the host controller must check each transaction before it starts to ensure that it completes before the end of the micro-frame.

So, what exactly needs to be involved in this check? Fundamentally, the transaction data payload, plus bit stuffing, plus transaction overhead must be taken into consideration. It is possible to be extremely accurate on how much time the next transaction takes. Take OUTs for an example. The host controller must fetch all of the OUT data from memory in order to send it onto the USB bus. A host controller implementation could pre-fetch all of the OUT data, and pre-compute the actual number of bits in the token and data packets. In addition, the system knows the depth of the target endpoint, so it could closely estimate turnaround time for handshake. In addition, the host controller knows the size of a handshake packet. Pre-computing effects of bit stuffing and summing up the other overhead numbers can allow the host controller to know exactly whether there is enough bus time, before EOF1 to complete the OUT transaction. To accomplish this particular approach takes an inordinate amount of time and hardware complexity.

The alternative is to make a reasonable guess whether the next transaction can be started. An example approximation algorithm is described below. This example algorithm relies on the EHCI policy that periodic transactions are scheduled first in the micro-frame. It is a reasonable assumption that software never over-commits the micro-frame to periodic transactions greater than the specification allowable 80%. In the available remaining 20% bandwidth, the host controller has some ability (in this example) to decide whether or not to execute a transaction. The result of this algorithm is that sometimes, under some circumstances a transaction is not executed that could have been executed. However, under all circumstances, a transaction is never started unless there is enough time in the frame to complete the transaction.

44.6.3.4.1.1 Transaction Fit - A Best-Fit Approximation Algorithm

A curve is calculated which represents the latest start time for every packet size, at which software schedules the start of a periodic transaction. This curve is the 80% bandwidth curve. Another curve is calculated which is the absolute, latest permitted start time for every packet size. This curve represents the absolute latest time, that a transaction of each packet size can be started and completed, in the micro-frame. A plot of these two curves are illustrated in Figure 44-146. The plot Y-axis represents the number of byte-times left in a frame.

The space between the 80% and the Last Start plots is bandwidth reclamation area. In this algorithm the host controller may skip transactions during this time if it is prudent.

The Best-Fit Approximation method plots a function (*f*(*x*)) between the 80% and Last Start curves. The function *f*(*x*) adds a constant to every transaction's maximum packet size and the result compared with the number of bytes left in the frame. The constant represents an approximation of the effects of bit stuffing and protocol overhead. The host controller starts transactions whose results land above the function curve. The host controller will not start transactions whose results land below the function curve.

Figure 44-146 illustrates the Best-Fit Approximation.

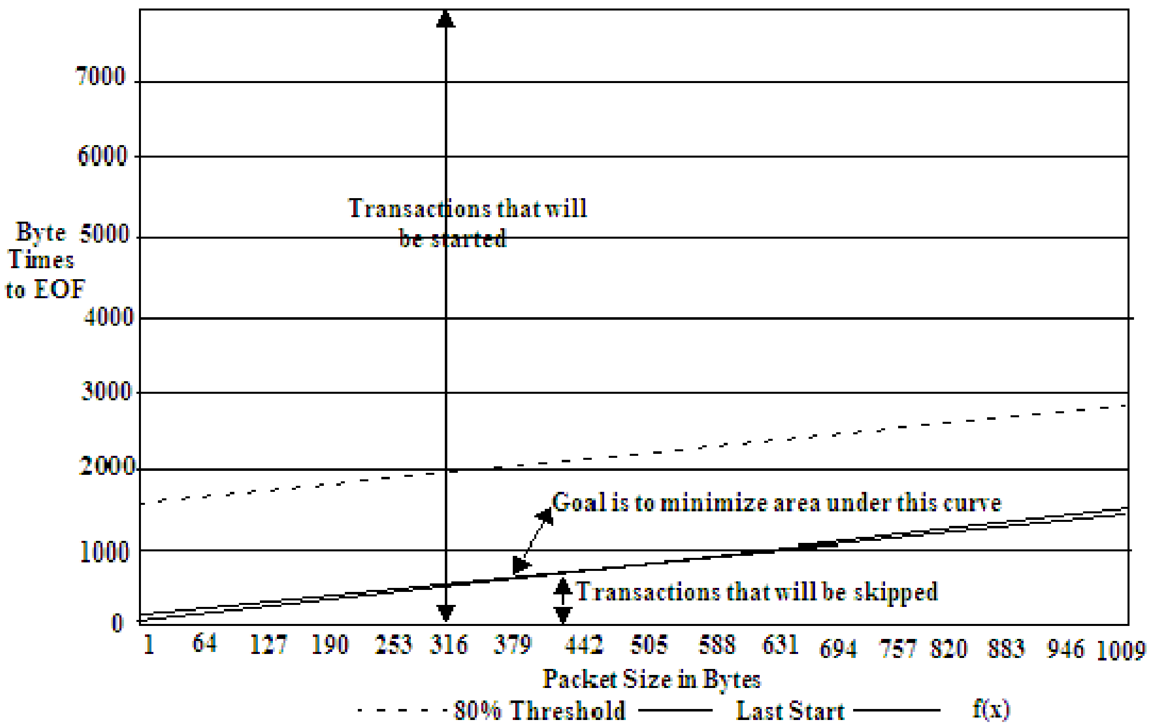


Figure 44-146. Best Fit Approximation

The LastStart line was calculated in this example to assume the absolute worst-case bus overhead per transaction. The particular transaction used is a start-split, zero-length OUT transaction with a handshake. Summaries of the component parts are listed in Table 44-181. The component times were derived from the protocol timings defined in the USB Specification Revision 2.0.

Table 44-181. Example Worse-case Transaction Timing Components

Component	Bit time	Byte Time	Explanation
Split Token	76	9.5	Split token as defined in USB core specification. Includes sync, token, eop, and so on.
Host 2 Host IPG	88	11	Number of bit times required between consecutive host packets.
Token	67	8.375	Token as defined in USB core specification. Includes sync, token, eop, and so on.
Host 2 Host IPG	88	11	Token as defined in USB core specification. Includessync, token, eop, and so on.
Data Packet (0 data bytes)	66.7	8.34	Zero-length data packet. Includes sync, PID, crc16, eop, and so on.
Turnaround time	721	90.125	Time for packet initiator (Host) to see the beginning of a response to a transmitted packet.
Handshake packet	48	6	Handshake packet as defined in USB core specification. Includes sync, PID, eop, and so on.
		144	Total

The exact details of the function (f(x)) are up to the particular implementation. However, it should be obvious that the goal is to minimize the area under the curve between the approximation function and the Last Start curve, without dipping below the LastStart line, while at the same time keeping the check as simple as possible for hardware implementation. The f(x) in Figure 44-146 was constructed using the following pseudo-code test on each transaction size data point. This algorithm assumes that the host controller keeps track of the remaining bits in the frame.

```
Alorithm CheckTransactionWillFit (MaximumPacketSize, HC_BytesLeftInFrame)
Begin
Local Temp = MaximumPacketSize + 192
Local rvalue = TRUE
If MaximumPacketSize >= 128 then
     Temp += 128
End If
If Temp > HC_BytesLeftInFrame then
     Rvalue = FALSE
End If
Return rvalue
End
```

This algorithm takes two inputs, the current maximum packet size of the transaction and the hardware counter of the number of bytes left in the current micro-frame. It unconditionally adds a simple constant of 192 to the maximum packet size to account for a first-order effect of transaction overhead and bit stuffing. If the transaction size is greater than or equal to 128 bytes, then an additional constant of 128 is added to the

running sum to account for the additional worst-case bit stuffing of payloads larger than 128. An inflection point was inserted at 128 because the *f*(*x*) plot was getting close to the LastStart line.

44.6.3.5 Periodic Schedule Frame Boundaries vs Bus Frame Boundaries

The USB Specification Revision 2.0 requires that the frame boundaries (SOF frame number changes) of the high-speed bus and the full- and low-speed bus(s) below USB 2.0 Hubs be strictly aligned. Super-imposed on this requirement is that USB 2.0 Hubs manage full- and low-speed transactions through a micro-frame pipeline (see start- (SS) and complete- (CS) splits illustrated in Figure 44-147). A simple, direct projection of the frame boundary model into the host controller interface schedule architecture creates tension (complexity for both hardware and software) between the frame boundaries and the scheduling mechanisms required to service the full- and low-speed transaction translator periodic pipelines.

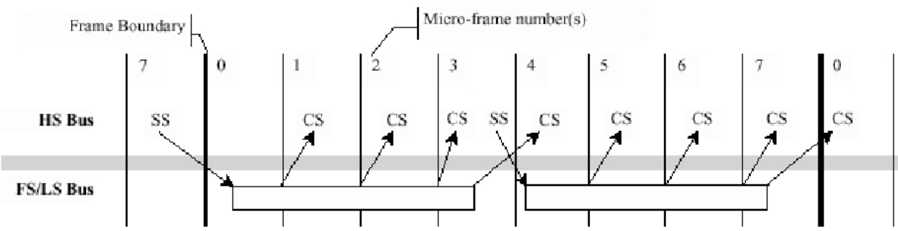


Figure 44-147. Frame Boundary Relationship between HS bus and FS/LS Bus

The simple projection, as Figure 44-147 illustrates, introduces frame-boundary wrap conditions for scheduling on both the beginning and end of a frame. In order to reduce the complexity for hardware and software, the host controller is required to implement one micro-frame phase shift for its view of frame boundaries. The phase shift eliminates the beginning of frame and frame-wrap scheduling boundary conditions.

The implementation of this phase shift requires that the host controller use one register value for accessing the periodic frame list and another value for the frame number value included in the SOF token. These two values are separate, but tightly coupled. The periodic frame list is accessed through the Frame List Index Register (USB_FRINDEX) documented in USB Frame Index (USB_FRINDEX) and initially illustrated in Schedule Traversal Rules . Bits FRINDEX[2:0], represent the micro-frame number. The SOF value is coupled to the value of FRINDEX[13:3]. Both FRINDEX[13:3] and the SOF value are increment based on FRINDEX[2:0]. It is required that the SOF value be delayed from the FRINDEX value by one micro-frame. The one micro-frame delay yields host controller periodic schedule and bus frame boundary relationship as illustrated in Figure 44-148.

This adjustment allows software to trivially schedule the periodic start and complete-split transactions for full-and low-speed periodic endpoints, using the natural alignment of the periodic schedule interface. The reasons for selecting this phase-shift are beyond the scope of this specification.

Figure 44-148 illustrates how periodic schedule data structures relate to schedule frame boundaries and bus frame boundaries. To aid the presentation, two terms are defined: The host controller's view of the 1 msec boundaries is called H-Frames. The high-speed bus's view of the 1 msec boundaries is called B-Frames.

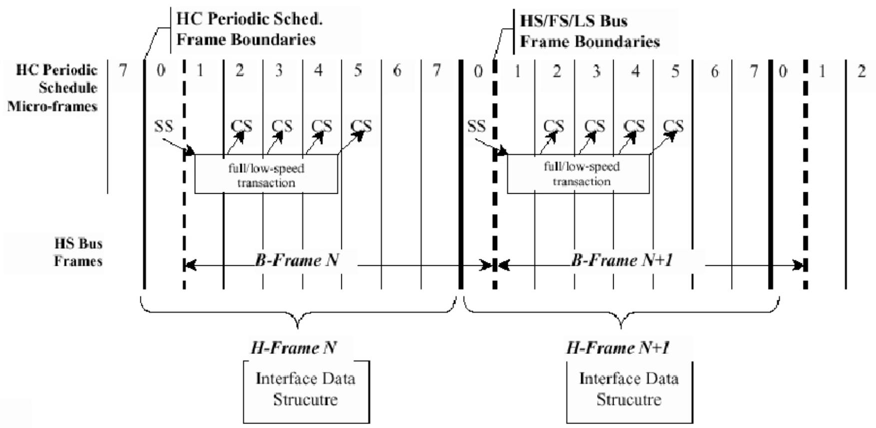


Figure 44-148. Relationship of Periodic Schedule Frame Boundaries to Bus Frame Boundaries

H-Frame boundaries for the host controller correspond to increments of FRINDEX[13:3]. Micro-frame numbers for the H-Frame are tracked by FRINDEX[2:0]. B-Frame boundaries are visible on the high-speed bus through changes in the SOF token's frame number. Micro-frame numbers on the high-speed bus are only derived from the SOF token's frame number (that is the high-speed bus sees eight SOFs with the same frame number value). H-Frames and B-Frames have the fixed relationship (that is B-Frames lag H-Frames by one micro-frame time) illustrated in Figure 44-148 . The host controller's periodic schedule is naturally aligned to H-Frames. Software schedules transactions for full- and low-speed periodic endpoints relative the H-Frames. The result is these transactions execute on the high-speed bus at exactly the right time for the USB 2.0 Hub periodic pipeline. As described in USB Frame Index (USB_FRINDEX), the SOF Value can be implemented as a shadow register (in this example, called SOFV), which lags the FRINDEX register bits [13:3] by one micro-frame count. This lag behavior can be accomplished by incrementing FRINDEX[13:3] based on carry-out on the 7 to 0 increment of FRINDEX[2:0] and incrementing SOFV based on the transition of 0 to 1 of FRINDEX[2:0].

Software is allowed to write to FRINDEX. USB Frame Index (USB_FRINDEX) provides the requirements that software should adhere when writing a new value in FRINDEX.

Table 44-182 illustrates the required relationship between the value of FRINDEX and the value of SOFV.

Table 44-182. Operation of FRINDEX and SOFV (SOF Value Register)

Current			Next		
FRINDEX[F]	SOFV	FRINDEX[mF]	FRINDEX[F]	SOFV	FRINDEX[mF]
N	N	111b	N+1	N	000b
N+1	N	000b	N+1	N+1	001b
N+1	N+1	001b	N+1	N+1	010b
N+1	N+1	010b	N+1	N+1	011b
N+1	N+1	011b	N+1	N+1	100b
N+1	N+1	100b	N+1	N+1	101b
N+1	N+1	101b	N+1	N+1	110b
N+1	N+1	110b	N+1	N+1	111b

NOTE

Where [F] = [13:3]; [μF] = [2:0]

44.6.3.6 Periodic Schedule

The periodic schedule traversal is enabled or disabled through the Periodic Schedule Enable bit in the USB_USBCMD register. If the Periodic Schedule Enable bit is set to zero, then the host controller simply does not try to access the periodic frame list through the USB_PERIODICLISTBASE register. Likewise, when the Periodic Schedule Enable bit is one, then the host controller does use the USB_PERIODICLISTBASE register to traverse the periodic schedule. The host controller will not react to modifications to the Periodic Schedule Enable immediately. In order to eliminate conflicts with split transactions, the host controller evaluates the Periodic Schedule Enable bit only when FRINDEX[2:0] is zero. System software must not disable the periodic schedule if the schedule contains an active split transaction work item that spans the 000b micro-frame. These work items must be removed from the schedule before the Periodic Schedule Enable bit is written to zero. The Periodic Schedule Status bit in the USB_USBSTS register indicates status of the periodic schedule. System software enables (or disables) the periodic schedule by writing one (or zero) to the Periodic Schedule Enable bit in the USB_USBCMD register. Software then can poll the Periodic Schedule Status bit to

determine when the periodic schedule has made the desired transition. Software must not modify the Periodic Schedule Enable bit unless the value of the Periodic Schedule Enable bit equals that of the Periodic Schedule Status bit.

The periodic schedule is used to manage all isochronous and interrupt transfer streams. The base of the periodic schedule is the periodic frame list. Software links schedule data structures to the periodic frame list to produce a graph of scheduled data structures. The graph represents an appropriate sequence of transactions on the

Figure 44-149 illustrates isochronous transfers (using iTDs and siTDs) with a period of one are linked directly to the periodic frame list. Interrupt transfers (are managed with queue heads) and isochronous streams with periods other than one are linked following the period-one iTD/siTDs. Interrupt queue heads are linked into the frame list ordered by poll rate. Longer poll rates are linked first (for example, closest to the periodic frame list), followed by shorter poll rates, with queue heads with a poll rate of one, on the very end.

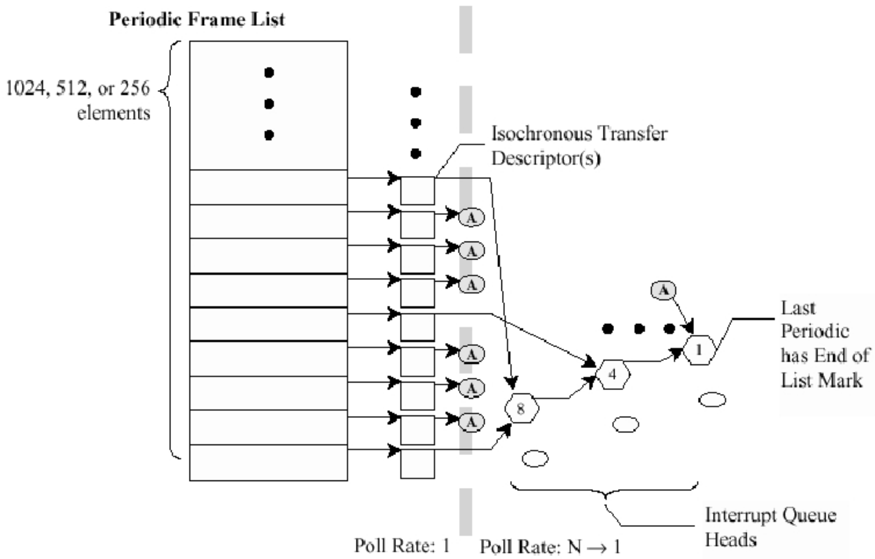


Figure 44-149. Example Periodic Schedule

44.6.3.7 Managing Isochronous Transfers Using iTDs

The structure of an iTD is presented in Isochronous (High-Speed) Transfer Descriptor (iTD). The four distinct sections to an iTD:

- The first field is the Next Link Pointer. This field is for schedule linkage purposes only.
- Transaction description array. This area is an eight-element array. Each element represents control and status information for one micro-frame's worth of transactions for a single high-speed isochronous endpoint.

- The buffer page pointer array is a 7-element array of physical memory pointers to data buffers. These are 4 K aligned pointers to physical memory.
- Endpoint capabilities. This area utilizes the unused low-order 12 bits of the buffer page pointer array. The fields in this area are used across all transactions executed for this iTD, including endpoint addressing, transfer direction, maximum packet size and high-bandwidth multiplier.

44.6.3.7.1 Host Controller Operational Model for iTDs

The host controller uses FRINDEX register bits [12:3] to index into the periodic frame list. This means that the host controller visits each frame list element eight consecutive times before incrementing to the next periodic frame list element. Each iTD contains eight transaction descriptions, which map directly to FRINDEX register bits [2:0]. Each iTD can span 8 micro-frames worth of transactions. When the host controller fetches an iTD, it uses FRINDEX register bits [2:0] to index into the transaction description array. If the active bit in the Status field of the indexed transaction description is set to zero, the host controller ignores the iTD and follows the Next pointer to the next schedule data structure.

When the indexed active bit is one, the host controller continues to parse the iTD. It stores the indexed transaction description and the general endpoint information (device address, endpoint number, maximum packet size, and so on.). It also uses the Page Select (PG) field to index the buffer pointer array, storing the selected buffer pointer and the next sequential buffer pointer. For example, if PG field is 0, then the host controller stores Page 0 and Page 1.

The host controller constructs a physical data buffer address by concatenating the current buffer pointer (as selected using the current transaction description's PG field) and the transaction description's Transaction Offset field. The host controller uses the endpoint addressing information and I/O-bit to execute a transaction to the appropriate endpoint. When the transaction is complete, the host controller clears the active bit and writes back any additional status information to the Status field in the currently selected transaction description.

The data buffer associated with the iTD must be virtually contiguous memory. Seven page pointers are provided to support eight high-bandwidth transactions regardless of the starting packet's offset alignment into the first page. A starting buffer pointer (physical memory address) is constructed by concatenating the page pointer (for example, page 0 pointer) selected by the active transaction descriptions' PG (for example, value: 00B) field with the transaction offset field. As the transaction moves data, the host controller must detect when an increment of the current buffer pointer crosses a page boundary. When this occurs the host controller simply replaces the current buffer pointer's page portion with the next page pointer (for example, page 1 pointer) and continues to move

data. The size of each bus transaction is determined by the value in the Maximum Packet Size field. An iTD supports high-bandwidth pipes through the Mult (multiplier) field. When the Mult field is 1, 2, or 3, the host controller executes the specified number of Maximum Packet sized bus transactions for the endpoint in the current micro-frame. In other words, the Mult field represents a transaction count for the endpoint in the current micro-frame. If the Mult field is zero, the operation of the host controller is undefined. The transfer description is used to service all transactions indicated by the Mult field.

For OUT transfers, the value of the Transaction X Length field represents the total bytes to be sent during the micro-frame. The Mult field must be set by software to be consistent with Transaction X Length and Maximum Packet Sixe. The host controller sends the bytes in Maximum Packet Size'd portions. After each transaction, the host controller decrements its local copy of Transaction X Length by Maximum Packet Size. The number of bytes the host controller sends is always Maximum Packet Size or Transaction X Length, whichever is less. The host controller advances the transfer state in the transfer description, updates the appropriate record in the iTD and moves to the next schedule data structure. The maximum sized transaction supported is 3 x 1024 bytes.

For IN transfers, the host controller issues Mult transactions. It is assumed that software has properly initialized the iTD to accommodate all of the possible data. During each IN transaction, the host controller must use Maximum Packet Size to detect packet babble errors. The host controller keeps the sum of bytes received in the Transaction X Length field. After all transactions for the endpoint have completed for the micro-frame, Transaction X Length contains the total bytes received. If the final value of Transaction X Length is less than the value of Maximum Packet Size, then less data than was allowed for was received from the associated endpoint. This short packet condition does not set the USBINT bit in the USB_USBSTS register to one. The host controller will not detect this condition. If the device sends more than Transaction X Length or Maximum Packet Size bytes (whichever is less), then the host controller sets the Babble Detected bit to one and set the Active bit to zero. Note, that the host controller is not required to update the iTD field Transaction X Length in this error scenario. If the Mult field is greater than one, then the host controller automatically executes the value of Mult transactions. The host controller will not execute all Mult transactions if:

- The endpoint is an OUT and Transaction X Length goes to zero before all the Mult transactions have executed (ran out of data), or
- The endpoint is an IN and the endpoint delivers a short packet, or an error occurs on a transaction before Mult transactions have been executed. The end of micro-frame may occur before all of the transaction opportunities have been executed. When this happens, the transfer state of the transfer description is advanced to reflect the progress that was made, the result written back to the iTD and the host controller proceeds to processing the next micro-frame. Refer to Appendix D for a table

summary of the host controller required behavior for all the high-bandwidth transaction cases.

44.6.3.7.2 Software Operational Model for iTDs

A client buffer request to an isochronous endpoint may span 1 to N micro-frames. When N is larger than one, system software may have to use multiple iTDs to read or write data with the buffer (if N is larger than eight, it must use more than one iTD).

Figure 44-150 illustrates the simple model of how a client buffer is mapped by system software to the periodic schedule (that is the periodic frame list and a set of iTDs). On the right is the client description of its request. The description includes a buffer base address plus additional annotations to identify which portions of the buffer should be used with each bus transaction. In the middle is the iTD data structures used by the system software to service the client request. Each iTD can be initialized to service up to 24 transactions, organized into eight groups of up to three transactions each. Each group maps to one micro-frame's worth of transactions. The EHCI controller does not provide per-transaction results within a micro-frame. It treats the per-micro-frame transactions as a single logical transfer. On the left is the host controller's frame list. System software establishes references from the appropriate locations in the frame list to each of the appropriate iTDs. If the buffer is large, then system software can use a small set of iTDs to service the entire buffer. System software can activate the transaction description records (contained in each iTD) in any pattern required for the particular data stream.

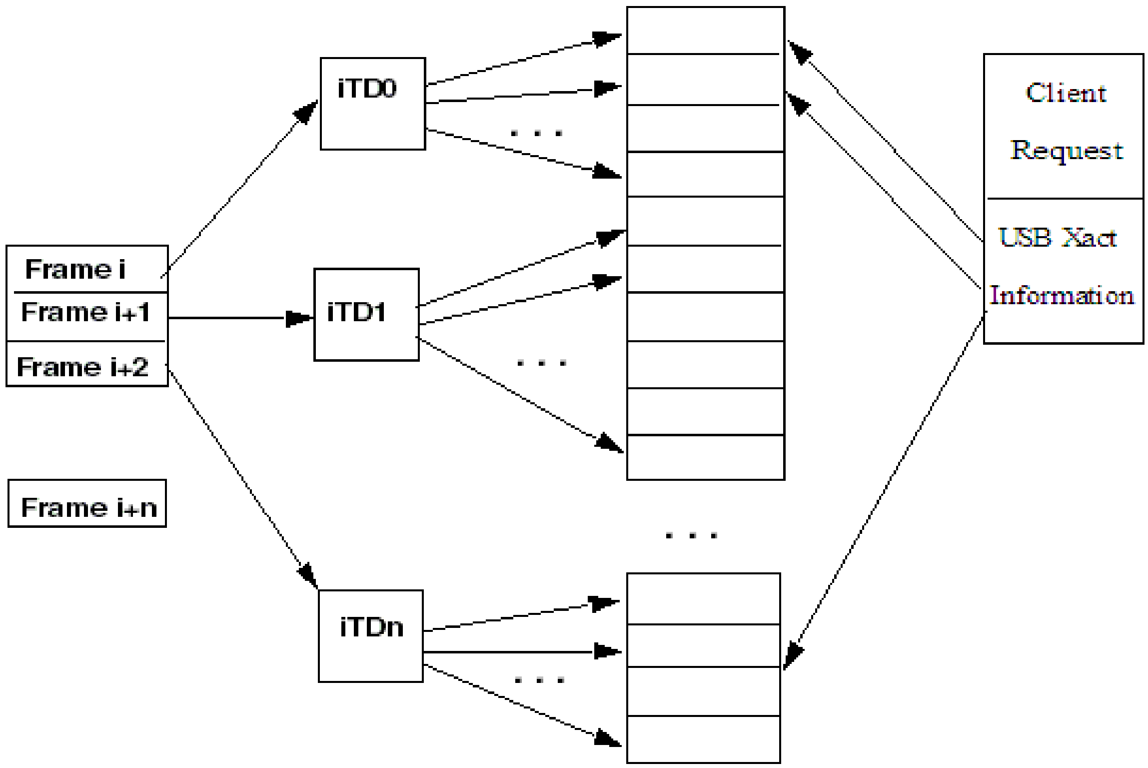


Figure 44-150. Example Association of iTDs to Client Request Buffer

As noted above, the client request includes a pointer to the base of the buffer and offsets into the buffer to annotate which buffer sections are to be used on each bus transaction that occurs on this endpoint. System software must initialize each transaction description in an iTD to ensure it uses the correct portion of the client buffer. For example, for each transaction description, the PG field is set to index the correct physical buffer page pointer and the Transaction Offset field is set relative to the correct buffer pointer page (for example, the same one referenced by the PG field). When the host controller executes a transaction it selects a transaction description record based on FRINDEX[2:0]. It then uses the current Page Buffer Pointer (as selected by the PG field) and concatenates to the transaction offset field. The result is a starting buffer address for the transaction. As the host controller moves data for the transaction, it must watch for a page wrap condition and properly advance to the next available Page Buffer Pointer. System software must not use the Page 6 buffer pointer in a transaction description where the length of the transfer wraps a page boundary. Doing so yields undefined behavior. The host controller hardware is not required to 'alias' the page selector to Page zero. USB 2.0 isochronous endpoints can specify a period greater than one. Software can achieve the appropriate scheduling by linking iTDs into the appropriate frames (relative to the frame list) and by setting appropriate transaction description elements active bits to one.

44.6.3.7.2.1 Periodic Scheduling Threshold

The Isochronous Scheduling Threshold field in the USB_HCCPARAMS capability register is an indicator to system software as to how the host controller pre-fetches and effectively caches schedule data structures. It is used by system software when adding isochronous work items to the periodic schedule. The value of this field indicates to system software the minimum distance it can update isochronous data (relative to the current location of the host controller execution in the periodic list) and still have the host controller process them.

The iTD and siTD data structures each describe 8 micro-frames worth of transactions. The host controller is allowed to cache one (or more) of these data structures in order to reduce memory traffic. Three basic caching models that account for the fact the isochronous data structures span 8 micro-frames. The three caching models are: no caching, micro-frame caching and frame caching.

When software is adding new isochronous transactions to the schedule, it always performs a read of the USB_FRINDEX register to determine the current frame and micro-frame the host controller is currently executing. Of course, there is no information about where in the micro-frame the host controller is, so a constant uncertainty-factor of one micro-frame has to be assumed. Combining the knowledge of where the host controller is executing with the knowledge of the caching model allows the definition of simple algorithms for how closely software can reliably work to the executing host controller.

No caching is indicated with a value of zero in the Isochronous Scheduling Threshold field. The host controller may pre-fetch data structures during a periodic schedule traversal (per micro-frame) but always dumps any accumulated schedule state at the end of the micro-frame. At the appropriate time relative to the beginning of every micro-frame, the host controller always begins schedule traversal from the frame list. Software can use the value of the USB_FRINDEX register (plus the constant 1 uncertainty-factor) to determine the approximate position of the executing host controller. When no caching is selected, software can add an isochronous transaction as near as 2 micro-frames in front of the current executing position of the host controller.

Frame caching is indicated with a non-zero value in bit [7] of the Isochronous Scheduling Threshold field. In the frame-caching model, system software assumes that the host controller caches one (or more) isochronous data structures for an entire frame (8 micro-frames). Software uses the value of the USB_FRINDEX register (plus the constant 1 uncertainty) to determine the current micro-frame/frame (assume modulo 8 arithmetic in adding the constant 1 to the micro-frame number). For any current frame N, if the current micro-frame is 0 to 6, then software can safely add isochronous transactions to Frame N + 1. If the current micro-frame is 7, then software can add isochronous transactions to Frame N + 2.

Micro-frame caching is indicated with a non-zero value in the least-significant 3 bits of the Isochronous Scheduling Threshold field. System software assumes the host controller caches one or more periodic data structures for the number of micro-frames indicated in the Isochronous Scheduling Threshold field. For example, if the count value were 2, then the host controller keeps a window of 2 micro-frames worth of state (current micro-frame, plus the next) on-chip. On each micro-frame boundary, the host controller releases the current micro-frame state and begins accumulating the next micro-frame state.

44.6.3.8 Asynchronous Schedule

The Asynchronous schedule traversal is enabled or disabled through the Asynchronous Schedule Enable bit in the USB_USBCMD register. If the Asynchronous Schedule Enable bit is set to zero, then the host controller simply does not try to access the asynchronous schedule through the USB_ASYNCLISTADDR register. Likewise, when the Asynchronous Schedule Enable bit is one, then the host controller does use the USB_ASYNCLISTADDR register to traverse the asynchronous schedule. Modifications to the Asynchronous Schedule Enable bit are not necessarily immediate. Rather the new value of the bit is taken into consideration the next time the host controller needs to use the value of the USB_ASYNCLISTADDR register to get the next queue head.

The Asynchronous Schedule Status bit in the USB_USBSTS register indicates status of the asynchronous schedule. System software enables (or disables) the asynchronous schedule by writing one (or zero) to the Asynchronous Schedule Enable bit in the USB_USBCMD register. Software then can poll the Asynchronous Schedule Status bit to determine when the asynchronous schedule has made the desired transition. Software must not modify the Asynchronous Schedule Enable bit unless the value of the Asynchronous Schedule Enable bit equals that of the Asynchronous Schedule Status bit.

The asynchronous schedule is used to manage all Control and Bulk transfers. Control and Bulk transfers are managed using queue head data structures. The asynchronous schedule is based at the USB_ASYNCLISTADDR register. The default value of the USB_ASYNCLISTADDR register after reset is undefined and the schedule is disabled when the Asynchronous Schedule Enable bit is zero.

Software may only write this register with defined results when the schedule is disabled. For example, Asynchronous Schedule Enable bit in the USB_USBCMD and the Asynchronous Schedule Status bit in the USB_USBSTS register are zero. System software enables execution from the asynchronous schedule by writing a valid memory address (of a queue head) into this register. Then software enables the asynchronous schedule by setting the Asynchronous Schedule Enable bit is set to one. The asynchronous schedule is actually enabled when the Asynchronous Schedule Status bit is one.

When the host controller begins servicing the asynchronous schedule, it begins by using the value of the USB_ASYNCLISTADDR register. It reads the first referenced data structure and begins executing transactions and traversing the linked list as appropriate. When the host controller completes processing the asynchronous schedule, it retains the value of the last accessed queue head's horizontal pointer in the USB_ASYNCLISTADDR register. Next time the asynchronous schedule is accessed, this is the first data structure that is serviced. This provides round-robin fairness for processing the asynchronous schedule.

A host controller completes processing the asynchronous schedule when one of the following events occur:

- The end of a micro-frame occurs.
- The host controller detects an empty list condition (see Empty Asynchronous Schedule Detection)
- The schedule has been disabled through the Asynchronous Schedule Enable bit in the USB_USBCMD register.

The queue heads in the asynchronous list are linked into a simple circular list as shown in Figure 44-145. Queue head data structures are the only valid data structures that may be linked into the asynchronous schedule. An isochronous transfer descriptor (iTD or siTD) in the asynchronous schedule yields undefined results.

The maximum packet size field in a queue head is sized to accommodate the use of this data structure for all non-isochronous transfer types. The USB Specification, Revision 2.0 specifies the maximum packet sizes for all transfer types and transfer speeds. System software should always parameterize the queue head data structures according to the core specification requirements.

44.6.3.8.1 Adding Queue Heads to Asynchronous Schedule

This is a software requirement section. There are two independent events for adding queue heads to the asynchronous schedule. The first is the initial activation of the asynchronous list. The second is inserting a new queue head into an activated asynchronous list.

Activation of the list is simple. System software writes the physical memory address of a queue head into the USB_ASYNCLISTADDR register, then enables the list by setting the Asynchronous Schedule Enable bit in the USB_USBCMD register to one.

When inserting a queue head into an active list, software must ensure that the schedule is always coherent from the host controllers' point of view. This means that the system software must ensure that all queue head pointer fields are valid. For example, qTD

pointers have T-Bits set to one or reference valid qTDs and the Horizontal Pointer references a valid queue head data structure. The following algorithm represents the functional requirements:

```
InsertQueueHead (pQHeadCurrent, pQueueHeadNew)
          --
          -- Requirement: all inputs must be properly initialized.
          --
          -- pQHeadCurrent is a pointer to a queue head that is
          -- already in the active list
          -- pQHeadNew is a pointer to the queue head to be added
          --
          -- This algorithm links a new queue head into a existing
          -- list
          --
          pQueueHeadNew.HorizontalPointer  =  pQueueHeadCurrent.HorizontalPointer
          pQueueHeadCurrent.HorizontalPointer =  physicalAddressOf(pQueueHeadNew)
End InsertQueueHead
```

44.6.3.8.2 Removing Queue Heads from Asynchronous Schedule

This is a software requirement section. There are two independent events for removing queue heads from the asynchronous schedule. The first is shutting down (deactivating) the asynchronous list. The second is extracting a single queue head from an activated list. Software deactivates the asynchronous schedule by setting the Asynchronous Schedule Enable bit in the USB_USBCMD register to zero. Software can determine when the list is idle when the Asynchronous Schedule Status bit in the USB_USBSTS register is zero. The normal mode of operation is that software removes queue heads from the asynchronous schedule without shutting it down. Software must not remove an active queue head from the schedule. Software should first deactivate all active qTDs, wait for the queue head to go inactive, then remove the queue head from the asynchronous list. Software removes a queue head from the asynchronous list through the following algorithm. As illustrated, the unlinking is quite easy. Software merely must ensure all of the link pointers reachable by the host controller are kept consistent.

```
UnlinkQueueHead (pQHeadPrevious, pQueueHeadToUnlink, pQHeadNext)
          --
          -- Requirement: all inputs must be properly initialized.
          --
          -- pQHeadPrevious is a pointer to a queue head that
          -- references the queue head to remove
          -- pQHeadToUnlink is a pointer to the queue head to be
          -- removed
          -- pQheadNext is a pointer to a queue head still in the
          -- schedule. Software provides this pointer with the
          -- following strict rules:
          --         if the host software is one queue head, then
          --         pQHeadNext must be the same as
          --         QueueheadToUnlink.HorizontalPointer. If the host
          --         software is unlinking a consecutive series of
          --         queue heads, QHeadNext must be set by software to
          --         the  queue head remaining in the schedule.
          --
          -- This algorithm unlinks a queue head from a circular list
          --
          pQueueHeadPrevious.HorizontalPointer =  pQueueHeadToUnlink.HorizontalPointer
```

```
                    pQueueHeadToUnlink.HorizontalPointer = pQHeadNext
End UnlinkQueueHead
```

If software removes the queue head with the H-bit set to one, it must select another queue head still linked into the schedule and set its H-bit to one. This should be completed before removing the queue head. The requirement is that software keep one queue head in the asynchronous schedule, with its H-bit set to one. At the point software has removed one or more queue heads from the asynchronous schedule, it is unknown whether the host controller has a cached pointer to them. Similarly, it is unknown how long the host controller might retain the cached information, as it is implementation dependent and may be affected by the actual dynamics of the schedule load. Therefore, once software has removed a queue head from the asynchronous list, it must retain the coherency of the queue head (link pointers, and so on). It cannot disturb the removed queue heads until it knows that the host controller does not have a local copy of a pointer to any of the removed data structures.

The method software uses to determine when it is safe to modify a removed queue head is to handshake with the host controller. The handshake mechanism allows software to remove items from the asynchronous schedule, then execute a simple, lightweight handshake that is used by software as a key that it can free (or reuse) the memory associated the data structures it has removed from the asynchronous schedule.

The handshake is implemented with three bits in the host controller. The first bit is a command bit (Interrupt on Async Advance Doorbell bit in the USB_USBCMD register) that allows software to inform the host controller that something has been removed from its asynchronous schedule. The second bit is a status bit (Interrupt on Async Advance bit in the USB_USBSTS register) that the host controller sets after it has released all on-chip state that may potentially reference one of the data structures just removed. When the host controller sets this status bit to one, it also sets the command bit to zero. The third bit is an interrupt enable (Interrupt on Async Advance bit in the USB_USBINTR register) that is matched with the status bit. If the status bit is one and the interrupt enable bit is one, then the host controller asserts a hardware interrupt.

Figure 44-151 illustrates a general example. In this example, consecutive queue heads (B and C) are unlinked from the schedule using the algorithm above. Before the unlink operation, the host controller has a copy of queue head A.

The unlink algorithm requires that as software unlinks each queue head, the unlinked queue head is loaded with the address of a queue head that remains in the asynchronous schedule.

When the host controller observes that doorbell bit being set to one, it makes a note of the local reachable schedule information. In this example, the local reachable schedule information includes both queue heads (A & B). It is sufficient that the host controller

can set the status bit (and clear the doorbell bit) as soon as it has traversed beyond current reachable schedule information (that is traversed beyond queue head (B) in this example). Figure 44-151 illustrates the generic queue head unlink scenario.

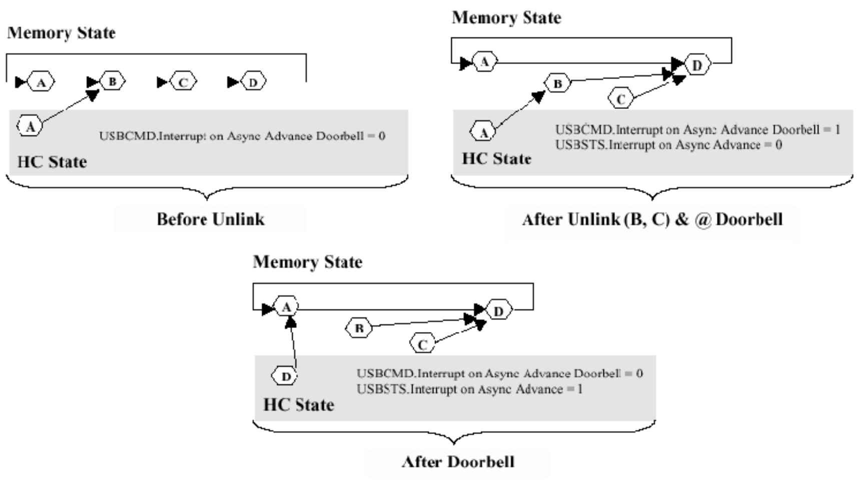


Figure 44-151. Generic Queue Head Unlink Scenario

Alternatively, a host controller implementation is allowed to traverse the entire asynchronous schedule list (for example, observed the head of the queue (twice)) before setting the Advance on Async status bit to one.

Software may re-use the memory associated with the removed queue heads after it observes the Interrupt on Async Advance status bit is set to one, following assertion of the doorbell. Software should acknowledge the Interrupt on Async Advance status as indicated in the USB_USBSTS register, before using the doorbell handshake again.

44.6.3.8.3 Empty Asynchronous Schedule Detection

The Enhanced Host Controller Interface uses two bits to detect when the asynchronous schedule is empty. The queue head data structure (see Table 44-168) defines an *H-bit* in the queue head, which allows software to mark a queue head as being the *head* of the reclaim list. The Enhanced Host Controller Interface also keeps a 1-bit flag in the USB_USBSTS register (*Reclamation*) that is set to zero when the Enhanced Interface Host Controller observes a queue head with the H-bit set to one. The reclamation flag in the status register is set to one when any USB transaction from the asynchronous schedule is executed (or whenever the asynchronous schedule starts, see Asynchronous Schedule Traversal: *Start Event*).

If the Enhanced Host Controller Interface ever encounters an *H-bit* of one and a *Reclamation* bit of zero, the EHCI controller simply stops traversal of the asynchronous schedule.

An example illustrating the H-bit in a schedule is shown in Figure 44-152

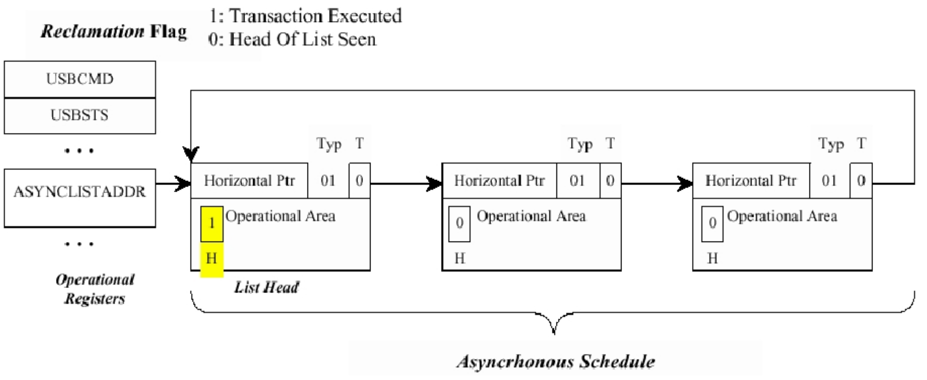


Figure 44-152. Asynchronous Schedule List w/Annotation to Mark Head of List

Software must ensure there is at most one queue head with the *H-bit* set to one, and that it is always coherent with respect to the schedule.

44.6.3.8.4 Restarting Asynchronous Schedule Before EOF

There are many situations where the host controller will detect an empty list *long* before the end of the micro-frame. It is important to remember that under many circumstances the schedule traversal has stopped due to Nak/Nyet responses from all endpoints.

An example of particular interest is when a start-split for a bulk endpoint occurs early in the micro-frame. Given the EHCI simple traversal rules, the complete-split for that transaction may Nak/Nyet out very quickly. If it is the only item in the schedule, then the host controller ceases traversal of the Asynchronous schedule very early in the micro-frame. In order to provide reasonable service to this endpoint, the host controller should issue the complete-split before the end of the current micro-frame, instead of waiting until the next micro-frame. When the reason for host controller idling asynchronous schedule traversal is because of empty list detection, it is mandatory the host controller implement a 'waking' method to resume traversal of the asynchronous schedule. An example method is described below.

44.6.3.8.4.1 Example Method for Restarting Asynchronous Schedule Traversal

The reason for idling the host controller when the list is empty is to keep the host controller from unnecessarily occupying too much memory bandwidth. The question is: *how long should the host controller stay idle before restarting*?

The answer in this example is based on deriving a manifest constant, which is the amount of time the host controller will stay idle before restarting traversal. In this example, the manifest constant is called *AsyncSchedSleepTime*, and has a value of 10 μsec. The value is derived based on the analysis in Example Derivation for *AsyncSchedSleepTime*, The traversal algorithm is simple:

- Traverse the Asynchronous schedule until the either an End-Of-micro-Frame event occurs, or an empty list is detected. If the event is an End-of-micro-Frame, go attempt to traverse the Periodic schedule. If the event is an empty list, then set a sleep timer and go to a *schedule sleep* state.
- When the sleep timer expires, set working context to the Asynchronous Schedule start condition and go to *schedule active* state. The start context allows the HC to reload *Nakcnt* fields, and so on. So the HC has a chance to run for more than one iteration through the schedule.

This process simply repeats itself each micro-frame. Figure 44-153 illustrates a sample state machine to manage the active and sleep states of the Asynchronous Schedule traversal policy. There are three states: Actively traversing the Asynchronous schedule, Sleeping, and Not Active. The last two are similar in terms of interaction with the Asynchronous schedule, but the Not Active state means that the host controller is busy with the Periodic schedule or the Asynchronous schedule is not enabled. The Sleeping state is specifically a special state where the host controller is just waiting for a period of time before resuming execution of the Asynchronous schedule.

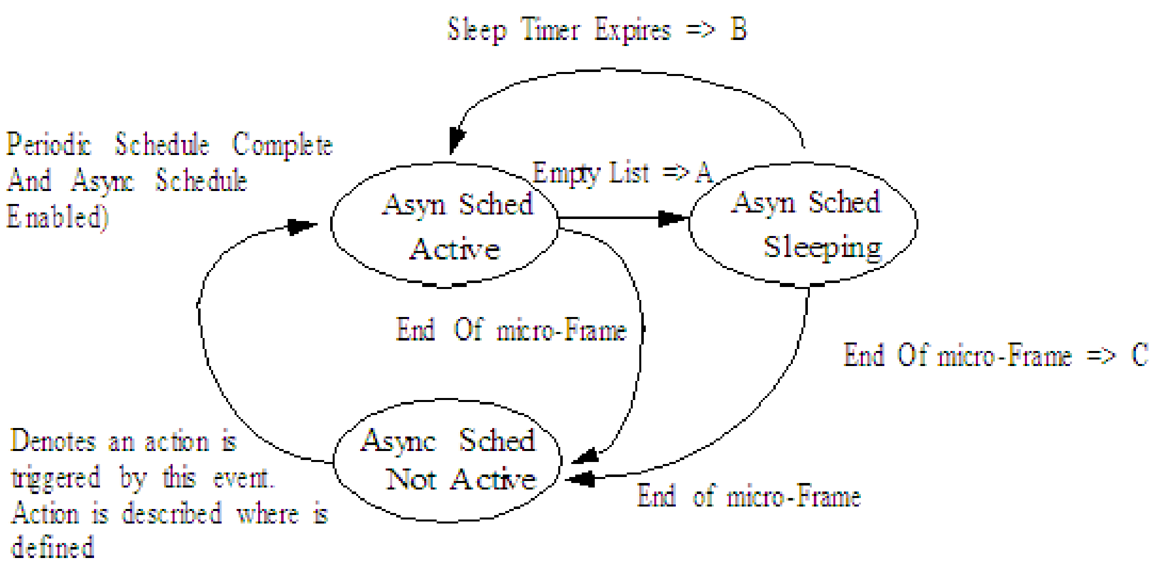


Figure 44-153. Example State Machine for Managing Asynchronous Schedule Traversal

The actions referred to in Figure 44-153 are defined in Table 44-183.

Table 44-183. Asynchronous Schedule SM Transition Actions

Action	Action Description Label
A	On detection of the empty list, the host controller sets the *AsynchronousTraversalSleepTimer* to *AsyncSchedSleepTime*.
B	When the *AsynchronousTraversalSleepTimer* expires, the host controller sets the *Reclamation* bit in the USBSTS register to one and moves the Nak Counter reload state machine to WaitForListHead (see Nak Count Reload Control).
C	The host controller cancels the sleep timer (*AsynchronousTraversalSleepTimer*).

44.6.3.8.4.2 Async Sched Not Active

This is the initial state of the traversal state machine after a host controller reset. The traversal state machine does not leave this state when the *Asynchronous Schedule Enable* bit in the USB_USBCMD register is zero.

This state is entered from Async Sched Active or Async Sched Sleeping states when the end-of-micro-frame event is detected.

44.6.3.8.4.3 Async Sched Active

This state is entered from the Async Sched Not Active state when the periodic schedule is not active. It is also entered from the Async Sched Sleeping states when the *AsyncrhonousTraversalSleepTimer* expires. On every transition into this state, the host controller sets the *Reclamation* bit in the USB_USBSTS register to one.

While in this state, the host controller continually traverses the asynchronous schedule until either the end of micro-frame or an empty list condition is detected.

44.6.3.8.4.4 Async Sched Sleeping

The state is entered from the Async Sched Active state when a schedule empty condition is detected. On entry to this state, the host controller sets the *AsynchronousTraversalSleepTimer* to *AsyncSchedSleepTime*.

44.6.3.8.4.5 Example Derivation for *AsyncSchedSleepTime*

The derivation is based on analysis of what work the host controller could be doing next. It assumes the host controller does not keep any state about what work is possibly pending in the asynchronous schedule. The schedule could contain any mix of the possible combinations of high- full- or low-speed control and bulk requests.

Table 44-184 summarizes some of the typical 'next transactions' that could be in the schedule, and the amount of time (for example *footprint*, or *wall clock*) the transaction takes to complete.

Table 44-184. Typical Low-/Full-speed Transaction Times

Transaction Attributes		Footprint (time)	Description
Speed	HS	11.9 ms	Maximum foot print for a worst-case, full-sized bulk data transaction.
Size	512	9.45 ms	Maximum footprint for an approximate best-case, full-sized bulk data transaction.
Type	Bulk		
Speed	FS	~50 ms	Approximate typical for full-sized bulk data. An 8-byte low-speed is about 2x, or between 90 and 100 ms.
Size	64		
Type	Bulk		
Speed	FS	~12 ms	Approximate typical for 8-byte bulk/control (that is setup)
Size	8		
Type	Cntrl		

A *AsyncSchedSleepTime* value of 10 μs provides a reasonable relaxation of the system memory load and still provides a good level of service for the various transfer types and payload sizes. For example, say we detect an empty list after issuing a start-split for a 64-byte full-speed bulk request. Assuming this is the only thing in the list, the host controller gets the results of the full-speed transaction from the hub during the fifth complete-split request. If the full-speed transaction was an IN and it nak'd, the 10 μs sleep period would allow the host controller to get the NAK results on the first complete-split.

44.6.3.8.5 Asynchronous Schedule Traversal: *Start Event*

Once the HC has *idled* itself through the empty schedule detection (Section 0), it will naturally *activate* and begin processing from the Periodic Schedule at the beginning of each micro-frame. In addition, it may have idled itself early in a micro-frame. When this occurs (idles early in the micro-frame) the HC must occasionally *re-activate* during the micro-frame and traverse the asynchronous schedule to determine whether any progress can be made. The requirements and method for this restart are described in Restarting Asynchronous Schedule Before EOF . Asynchronous schedule *Start Event*s are defined to be:

- Whenever the host controller transitions from the periodic schedule to the asynchronous schedule. If the periodic schedule is disabled and the asynchronous

schedule is enabled, then the beginning of the micro-frame is equivalent to the transition from the periodic schedule, or

- The asynchronous schedule traversal restarts from a sleeping state (see Restarting Asynchronous Schedule Before EOF).

44.6.3.8.6 Reclamation Status Bit (USBSTS Register)

The operation of the empty asynchronous schedule detection feature (see Empty Asynchronous Schedule Detection) depends on the proper management of the *Reclamation* bit in the USB_USBSTS register. The host controller tests for an empty schedule just after it fetches a new queue head while traversing the asynchronous schedule (see Fetch Queue Head).

It is required that the host controller sets the *Reclamation* bit to one whenever an asynchronous schedule traversal *Start Event*, as documented in Asynchronous Schedule Traversal: *Start Event*, occurs. The *Reclamation* bit is also set to one whenever the host controller executes a transaction while traversing the asynchronous schedule (see Execute Transaction). The host controller sets the *Reclamation* bit to zero whenever it finds a queue head with its *H-bit* set to one. Software should only set a queue head's *H-bit* if the queue head is in the asynchronous schedule. If software sets the *H-bit* in an interrupt queue head to one, the resulting behavior is undefined. The host controller may set the *Reclamation* bit to zero when executing from the periodic schedule.

44.6.3.9 Operational Model for Nak Counter

This section describes the operational model for the *NakCnt* field defined in a queue head (see Queue Head Initialization). Software should not use this feature for interrupt queue heads. This rule is not required to be enforced by the host controller.

USB protocol has built-in flow control through the Nak response by a device. There are several scenarios, beyond the Ping feature, where an endpoint may naturally Nak or Nyet the majority of the time. An example is the host controller management of the split transaction protocol for control and bulk endpoints. All bulk endpoints (High- or Full-speed) are serviced through the same asynchronous schedule. The time between the *Start-split* transaction and the first *Complete-split* transaction could be very short (that is like when the endpoint is the only one in the asynchronous schedule). The hub NYETs (effectively Naks) the *Complete-split* transaction until the classic transaction is complete. This could result in the host controller thrashing memory, repeatedly fetching the queue head and executing the transaction to the Hub, which does not complete until after the transaction on the classic bus completes.

The two component fields in a queue head to support the throttling feature: a counter field (*NakCnt*), and a counter reload field (*RL*). *NakCnt* is used by the host controller as one of the criteria to determine whether or not to execute a transaction to the endpoint. The two operational modes associated with this counter:

- Not Used- This mode is set when the *RL* field is zero. The host controller ignores the *NakCnt* field for any execution of transactions through a queue head with an *RL* field of zero. Software must use this selection for interrupt endpoints.
- Nak Throttle Mode- This mode is selected when the *RL* field is non-zero. In this mode, the value in the *NakCnt* field represents the maximum number of Nak or Nyet responses the host controller tolerates on each endpoint. In this mode, the HC decrements the *NakCnt* field based on the token/handshake criteria listed in Table 44-185 The host controller must reload *NakCnt* when the endpoint successfully moves data (for example, policy to reward device for moving data).

Table 44-185 describes the NakCnt field adjustment rules.

Table 44-185. NakCnt Field Adjustment Rules

Token	Handshake	
	Handshake NAK	NYET
IN/PING	decrement *NakCnt*	N/A (protocol error)
OUT	decrement *NakCnt*	No Action[1, 1] Start
Split	decrement *NakCnt*	N/A (protocol error)
Complete Split	No Action	Decrement *NakCnt*

1. Recommended behavior on this response is to reload *NakCnt*

In summary, system software enables the counter by setting the reload field (*RL*) to a non-zero value. The host controller may execute a transaction if *NakCnt* is non-zero. The host controller does not execute a transaction if *NakCnt* is zero. The reload mechanism is described in detail in Nak Count Reload Control .

NOTE

When all queue heads in the Asynchronous Schedule either exhausts all transfers or all NakCnt's go to zero, then the host controller detects an empty Asynchronous Schedule and idle schedule traversal (see Empty Asynchronous Schedule Detection).

Any time the host controller begins a new traversal of the Asynchronous Schedule, a *Start Event* is assumed, see Asynchronous Schedule Traversal: *Start Event*. Every time a Start-Event occurs, the Nak Count reload procedure is enabled.

44.6.3.9.1 Nak Count Reload Control

When the host controller reaches the *Execute Transaction* state for a queue head (meaning that it has an active operational state), it checks to determine whether the *NakCnt* field should be reloaded from *RL* (see Execute Transaction). If the answer is yes, then *RL* is copied into *NakCnt*. After the reload or if the reload is not active, the host controller evaluates whether to execute the transaction.

The host controller must reload nak counters (*NakCnt* see Table 44-168) in queue heads during the first pass through the reclamation list after an asynchronous schedule Start Event (see Asynchronous Schedule Traversal: *Start Event* for the definition of the Start Event). The Asynchronous Schedule should have at most one queue head marked as the head (see Figure 44-152).

Figure 44-154 illustrates an example state machine that satisfies the operational requirements of the host controller detecting the first pass through the Asynchronous Schedule. This state machine is maintained internal to the host controller and is only used to gate reloading of the nak counter during the queue head traversal state: Execute Transaction (Figure 44-154). The host controller does not perform the nak counter reload operation if the RL field (see Table 44-168) is set to zero.

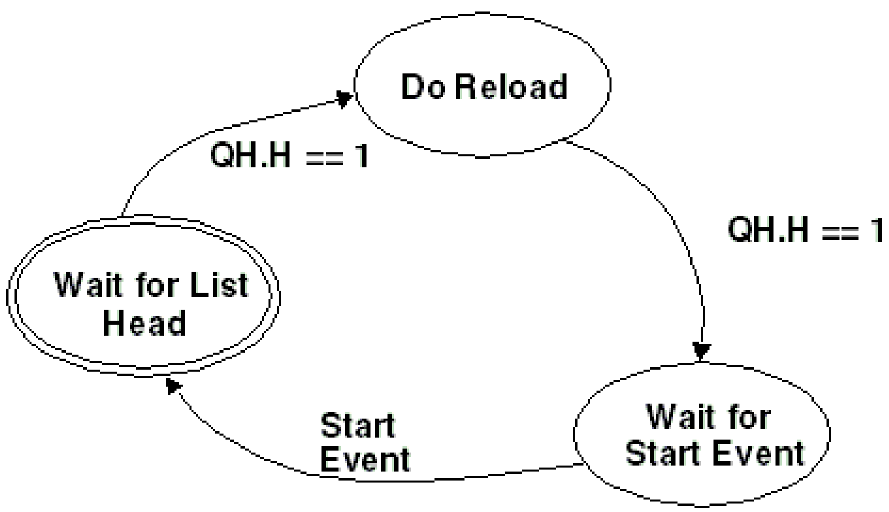


Figure 44-154. Example HC State Machine for Controlling Nak Counter Reloads

44.6.3.9.1.1 Wait for List Head

This is the initial state. The state machine enters this state from Wait for Start Event when a start event as defined in Asynchronous Schedule Traversal: *Start Event* occurs. The purpose of this state is to wait for the first observation of the head of the Asynchronous Schedule. This occurs when the host controller fetches a queue head whose *H-bit* is set to one.

44.6.3.9.1.2 Do Reload

This state is entered from the Wait for List Head state when the host controller fetches a queue head with the *H-bit* set to one. While in this state, the host controller performs nak counter reloads for every queue head visited that has a non-zero nak reload value (*RL*) field.

44.6.3.9.1.3 Wait for Start Event

This state is entered from the *Do Reload* state when a queue head with the *H-bit* set to one is fetched. While in this state, the host controller does not perform nak counter reloads.

44.6.3.10 Managing Control/Bulk/Interrupt Transfers through Queue Heads

This section presents an overview of how the host controller interacts with queuing data structures.

Queue heads use the Queue Element Transfer Descriptor (qTD) structure. One queue head is used to manage the data stream for one endpoint. The queue head structure contains static endpoint characteristics and capabilities. It also contains a working area from where individual bus transactions for an endpoint are executed (see Overlay area defined in Table 44-168). Each qTD represents one or more bus transactions, which is defined in the context of this specification as a *transfer*.

The general processing model for the host controller's use of a queue head is simple:

- read a queue head,
- execute a transaction from the overlay area,
- write back the results of the transaction to the overlay area,
- move to the next queue head.

If the host controller encounters errors during a transaction, the host controller sets one (or more) of the error reporting bits in the queue head's *Status* field. The *Status* field accumulates all errors encountered during the execution of a qTD (for example, the error bits in the queue head *Status* field are 'sticky' until the transfer (qTD) has completed). This state is always written back to the source qTD when the transfer is complete. On transfer (for example, buffer or halt conditions) boundaries, the host controller must auto-advance (without software intervention) to the next qTD. Additionally, the hardware must be able to halt the queue so no additional bus transactions occurs for the endpoint and the host controller does not advance the queue.

An example host controller operational state machine of a queue head traversal is illustrated in Figure 44-155. This state machine is a model for how a host controller should traverse a queue head. The host controller must be able to advance the queue from the *Fetch QH* state in order to avoid all hardware/software race conditions. This simple mechanism allows software to simply link qTDs to the queue head and *activate* them, then the host controller always *find* them if/when they are reachable. Figure 44-155 illustrates the Host Controller Queue Head Traversal State Machine.

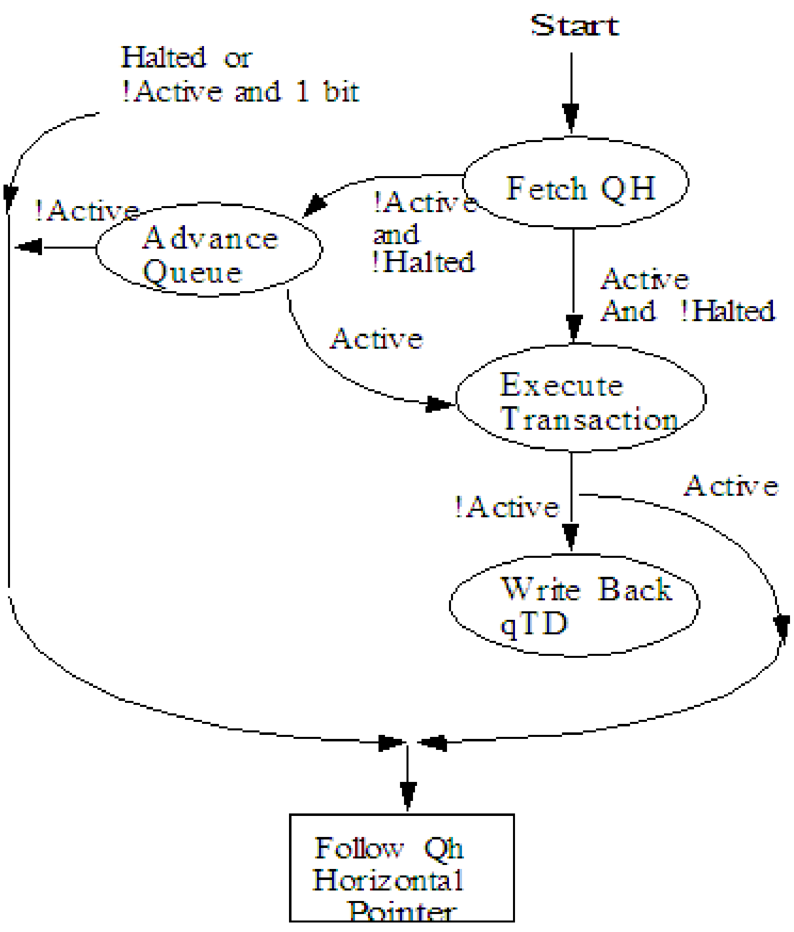


Figure 44-155. Host Controller Queue Head Traversal State Machine

This traversal state machine applies to all queue heads, regardless of transfer type or whether split transactions are required. The following sections describe each state. Each state description describes the entry criteria. The Execute Transaction state (see Execute Transaction) describes the basic requirements for all endpoints. Split Transactions for Asynchronous Transfers and Split Transaction Interrupt describe details of the required extensions to the Execute Transaction state for endpoints requiring split transactions.

NOTE

Prior to software placing a queue head into either the periodic or asynchronous list, software must ensure the queue head is properly initialized. Minimally, the queue head should be initialized to the following (see Section Queue Head for layout of a queue head):

Valid static endpoint state.

- For the very first use of a queue head, software may zero-out the queue head transfer overlay, then set the *Next qTD Pointer* field value to reference a valid qTD.

44.6.3.10.1 Fetch Queue Head

A queue head can be referenced from the physical address stored in the ASYNCLISTADDR Register (see Next Asynch. Address (USB_ASYNCLISTADDR))/ Endpoint List Address (USB_ENDPTLISTADDR) Additionally, it may be referenced from the *Next LinkPointer* field of an iTD, siTD, FSTN or another Queue Head. If the referencing link pointer has the *Typ* field set to indicate a queue head, it is assumed to reference a queue head structure as defined in Table 44-168.

While in this state, the host controller performs operations to implement empty schedule detection (see Empty Asynchronous Schedule Detection) and Nak Counter reloads (see Operational Model for Nak Counter). After the queue head has been fetched, the host controller conducts the following queries for empty schedule detection:

- If queue head is not an interrupt queue head (that is *S-mask* is zero), and
- The *H-bit* is one, and
- The *Reclamation* bit in the USBSTS register is zero.

When these criteria are met, the host controller stops traversing the asynchronous list (as described in Empty Asynchronous Schedule Detection). When the criteria are not met, the host controller continues schedule traversal. If the queue head is not an interrupt and the *H-bit* is one and the *Reclamation* bit is one, then the host controller sets the *Reclamation* bit in the USBSTS register to zero before completing this state. The operations for reloading of the Nak Counter are described in detail in Operational Model for Nak Counter.

This state is complete when the queue head has been read on-chip.

44.6.3.10.2 Advance Queue

To advance the queue, the host controller must find the next qTD, adjust pointers, perform the overlay and write back the results to the queue head.

This state is entered from the FetchQHD state if the overlay *Active* and *Halt* bits are set to zero. On entry to this state, the host controller determines which next pointer to use to fetch a qTD, fetches a qTD and determines whether or not to perform an overlay.

NOTE

If the *I-bit* is one and the *Active* bit is zero, the host controller immediately skips processing of this queue head, exits this state and uses the horizontal pointer to the next schedule data structure. If the field *Bytes to Transfer* is not zero and the *T-bit* in the *Alternate Next qTD Pointer* is set to zero, then the host controller uses the *Alternate Next qTD Pointer*. Otherwise, the host controller uses the *NextqTD Pointer*. If *NextqTD Pointer's T-bit* is set to one, then the host controller exits this state and uses the horizontal pointer to the next schedule data structure.

Using the selected pointer the host controller fetches the referenced qTD. If the fetched qTD has its *Active* bit set to one, the host controller moves the pointer value used to reach the qTD (*Next* or *Alternate Next*) to the *Current qTD Pointer* field, then performs the overlay. If the fetched qTD has its *Active* bit set to zero, the host controller aborts the queue advance and follows the queue head's horizontal pointer to the next schedule data structure.

The host controller performs the overlay based on the following rules:

- The value of the data toggle (*dt*) field in the overlay area depends on the value of the *data toggle control* (*dtc*) bit (see Table 44-170).
- If the *EPS* field indicates the endpoint is a high-speed endpoint, the *Ping* state field is preserved by the host controller. The value of this field is not changed as a result of the overlay.
- *C-prog-mask* field is set to zero (field from incoming qTD is ignored, as is the current contents of the overlay area).
- *Frame Tag* field is set to zero (field from incoming qTD is ignored, as is the current contents of the overlay area).
- *NakCnt* field in the overlay area is loaded from the *RL* field in the queue head's Static Endpoint State.
- All other areas of the overlay are set by the incoming qTD.

The host controller exits this state when it has committed the write to the queue head.

44.6.3.10.3 Execute Transaction

The host controller enters this state from the Fetch Queue Head state only if the *Active* bit in *Status* field of the queue head is set to one.

On entry to this state, the host controller executes a few pre-operations, then checks some pre-condition criteria before committing to executing a transaction for the queue head.

The pre-operations performed and the pre-condition criteria depend on whether the queue head is an interrupt endpoint. The host controller can determine that a queue head is an interrupt queue head when the queue head's *S-mask* field contains a non-zero value. It is the responsibility of software to ensure the *S-mask* field is appropriately initialized based on the transfer type. There are other criteria that must be met if the *EPS* field indicates that the endpoint is a low- or full-speed endpoint, see Split Transactions for Asynchronous Transfers and Split Transaction Interrupt .

44.6.3.10.3.1 Interrupt Transfer Pre-condition Criteria

If the queue head is for an interrupt endpoint (for example, non-zero *S-mask* field), then the FRINDEX[2:0] field must identify a bit in the *S-mask* field that has one in it. For example, an *S-mask* value of 00100000b would evaluate to true only when FRINDEX[2:0] is equal to 101b. If this condition is met then the host controller considers this queue head for a transaction.

44.6.3.10.3.2 Asynchronous Transfer Pre-operations and Pre-condition Criteria

If the queue head is not for an interrupt endpoint (for example, zero *S-mask* field), then the host controller performs one pre-operation and then evaluates one pre-condition criteria:

The pre-operation is:

Checks the Nak counter reload state (Operational Model for Nak Counter). It may be necessary for the host controller to reload the Nak Counter field. The reload is performed at this time.

The pre-condition evaluated is:

- Whether or not the *NakCnt* field has been reloaded, the host controller checks the value of the *NakCnt* field in the queue head. If *NakCnt* is non-zero, or if the *Reload Nak Counter* field is zero, then the host controller considers this queue head for a transaction.

44.6.3.10.3.3 Transfer Type Independent Pre-operations

Regardless of the transfer type, the host controller always performs at least one pre-operation and evaluates one pre-condition. The pre-operation is:

- A host controller internal transaction (down) counter *qHTransactionCounter* is loaded from the queue head's *Mult* field. A host controller implementation is allowed to ignore this for queue heads on the asynchronous list. It is mandatory for interrupt queue heads. Software should ensure that the *Mult* field is set appropriately for the transfer type.

The pre-conditions evaluated are:

- The host controller determines whether there is enough time in the micro-frame to complete this transaction (see Transaction Fit - A Best-Fit Approximation Algorithm for an example evaluation method). If there is not enough time to complete the transaction, the host controller exits this state.
- If the value of *qHTransactionCounter* for an interrupt endpoint is zero, then the host controller exits this state.

When the pre-operations are complete and pre-conditions are met, the host controller sets the *Reclamation* bit in the USBSTS register to one and then begins executing one or more transactions using the endpoint information in the queue head. The host controller iterates *qHTransactionCounter* times in this state executing transactions. After each transaction is executed, *qHTransactionCounter* is decremented by one. The host controller exits this state when one of the following events occurs:

- The *qHTransactionCounter* decrements to zero, or
- The endpoint responds to the transaction with any handshake other than an ACK,[4] or
- The transaction experiences a transaction error, or
- The *Active* bit in the queue head goes to zero, or
- There is not enough time in the micro-frame left to execute the next transaction (see Transaction Fit - A Best-Fit Approximation Algorithm for example method for implementing the frame boundary test).

NOTE

For a high-bandwidth interrupt OUT endpoint, the host controller may optionally immediately retry the transaction if it fails.

The results of each transaction is recorded in the on-chip overlay area. If data was successfully moved during the transaction, the transfer state in the overlay area is advanced. To advance queue head's transfer state, the *Total Bytes to Transfer* field is decremented by the number of bytes moved in the transaction, the data toggle bit (*dt*) is

toggled, the current page offset is advanced to the next appropriate value (for example, advanced by the number of bytes successfully moved), and the *C_Page* field is updated to the appropriate value (if necessary). See Buffer Pointer List Use for Data Streaming with qTDs .

NOTE

The *Total Bytes To Transfer* field may be zero when all the other criteria for executing a transaction are met. When this occurs, the host controller executes zero-length transaction to the endpoint. If the *PID_Code* field indicates an IN transaction and the device delivers data, the host controller detects a packet babble condition, set the *babble* and *halted* bits in the *Status* field, set the *Active* bit to zero, write back the results to the source qTD, then exit this state.

In the event an IN token receives a data PID mismatch response, the host controller must ignore the received data (for example not advance the transfer state for the bytes received). Additionally, if the endpoint is an interrupt IN, then the host controller must record that the transaction occurred (for example, decrement *qHTransactionCounter*). It is recommended (but not required) the host controller continue executing transactions for this endpoint if the resultant value of *qHTransactionCounter* is greater than one.

If the response to the IN bus transaction is a Nak (or Nyet) and *RL* is non-zero, *NakCnt* is decremented by one. If *RL* is zero, then no write-back by the host controller is required (for a transaction receiving a Nak or Nyet response and the value of *CErr* did not change). Software should set the *RL* field to zero if the queue head is an interrupt endpoint. Host controller hardware is not required to enforce this rule or operation.

After the transaction has finished and the host controller has completed the post processing of the results (advancing the transfer state and possibly *NakCnt*, the host controller writes back the results of the transaction to the queue head's overlay area in main memory).

The number of bytes moved during an IN transaction depends on how much data the device endpoint delivers. The maximum number of bytes a device can send is *MaximumPacket Size*. The number of bytes moved during an OUT transaction is either *Maximum Packet Length* bytes or *Total Bytes to Transfer*, whichever is less.

If there was a transaction error during the transaction, the transfer state (as defined above) is not advanced by the host controller. The *CErr* field is decremented by one and the status field is updated to reflect the type of error observed. Transaction errors are summarized in Transaction Error .

The following events causes the host controller to clear the *Active* bit in the queue head's overlay status field. When the *Active* bit transitions from one to zero, the transfer in the overlay is considered complete. The reason for the transfer completion (clearing the *Active* bit) determines the next state.

- *CErr* field decrements to zero. When this occurs the *Halted* bit is set to one and *Active* is set to zero. This results in the hardware not advancing the queue and the pipe halts. Software must intercede to recover.
- The device responds to the transaction with a STALL PID. When this occurs, the *Halted* bit is set to one and the *Active* bit is set to zero. This results in the hardware not advancing the queue and the pipe halts. Software must intercede to recover.
- The *Total Bytes to Transfer* field is zero after the transaction completes.
  - For a zero length transaction, it was zero before the transaction was started. When this condition occurs, the *Active* bit is set to zero.
- The PID code is an IN, and the number of bytes moved during the transaction is less than the *Maximum Packet Length*. When this occurs, the *Active* bit is set to zero and a short packet condition exists. The short-packet condition is detected during the Advance Queue state. Refer to Split Transactions for additional rules for managing low- and full-speed transactions.

With the exception of a NAK response (when *RL* field is zero), the host controller always writes the results of the transaction back to the overlay area in main memory. This includes when the transfer completes. For a high-speed endpoint, the queue head information written back includes minimally the following fields: The *PID Code* field indicates an IN and the device sends more than the expected number of bytes (for example *Maximum Packet Length* or *Total Bytes to Transfer* bytes, whichever is less) (for example a packet babble). This results in the host controller setting the *Halted* bit to one.

- NakCnt, dt, Total Bytes to Transfer, C_Page, Status, CERR, and Current Offset

For a low- or full-speed device the queue head information written back also includes the fields:

- C-prog-mask, FrameTag and S-bytes.

The duration of this state depends on the time it takes to complete the transaction(s) and the status write to the overlay is committed.

44.6.3.10.3.4 Halting a Queue Head

A halted endpoint is defined only for the transfer types that are managed through queue heads (control, bulk and interrupt). The following events indicate that the endpoint has reached a condition where no more activity can occur without intervention from the driver:

- An endpoint may return a STALL handshake during a transaction,
- A transaction had three consecutive error conditions, or
- A Packet Babble error occurs on the endpoint.

When any of these events occur (for a queue head) the Host Controller halts the queue head and set the USBERRINT status bit in the USB.USBSTS register to one. To halt the queue head, the *Active* bit is set to zero and the *Halted* bit is set to one. There may be other error status bits that are set when a queue is halted. The host controller always writes back the overlay area to the source qTD when the transfer is complete, regardless of the reason (normal completion, short packet or halt). The host controller does not advance the transfer state on a transaction that results in a *Halt* condition (for example no updates necessary for *Total Bytes to Transfer*, *C_Page*, *Current Offset,* and *dt*). The host controller must update *CErr* as appropriate. When a queue head is halted, the *USB Error Interrupt* bit in the USB.USBSTS register is set to one. If the *USB Error Interrupt Enable* bit in the USB.USBINTR register is set to one, a hardware interrupt is generated at the next interrupt threshold.

44.6.3.10.3.5 Asynchronous Schedule Park Mode

Asynchronous Schedule Park mode is a special execution mode that can be enabled by system software, where the host controller is permitted to execute more than one bus transaction from a high-speed queue head in the Asynchronous schedule before continuing horizontal traversal of the Asynchronous schedule. This feature has no effect on queue heads or other data structures in the Periodic schedule. This feature is similar in intent as the *Mult* feature that is used in the Periodic schedule. Where-as the *Mult* feature is a characteristic that is tunable for each endpoint; park-mode is a policy that is applied to all high-speed queue heads in the asynchronous schedule. It is essentially the specification of an iterator for consecutive bus transactions to the same endpoint. All of the rules for managing bus transactions and the results of those as defined in Execute Transaction apply. This feature merely specifies how many consecutive times the host controller is permitted to execute from the same queue head before moving to the next queue head in the Asynchronous List. This feature should allow the host controller to attain better bus utilization for those devices that are capable of moving data at maximum rate, while at the same time providing a fair service to all endpoints.

A host controller exports its capability to support this feature to system software by setting the *Asynchronous Schedule Park Capability* bit in the USB.HCCPARAMs register to one. This information keys system software that the *Asynchronous Schedule Park Mode Enable* and *Asynchronous Schedule Park Mode Count* fields in the USB.USBCMD register are modifiable. System software enables the feature by writing a one to the *Asynchronous Schedule Park Mode Enable* bit.

When park-mode is not enabled (for example *Asynchronous Schedule Park Mode Enable* bit in the USB.USBCMD register is zero), the host controller must not execute more than one bus transaction per high-speed queue head, per traversal of the asynchronous schedule. When park-mode is enabled, the host controller must not apply the feature to a queue head whose *EPS* field indicates a Low/Full-speed device (for example only one bus transaction is allowed from each Low/Full-speed queue head per traversal of the asynchronous schedule). Park-mode may only be applied to queue heads in the Asynchronous schedule whose *EPS* field indicates that it is a high-speed device.

The host controller must apply park mode to queue heads whose *EPS* field indicates a high-speed endpoint. The maximum number of consecutive bus transactions a host controller may execute on a high-speed queue head is determined by the value in the *Asynchronous Schedule Park Mode Count* field in the USB.USBCMD register. Software must not set *Asynchronous Schedule Park Mode Enable* bit to one and also set *Asynchronous Schedule Park Mode Count* field to zero. The resulting behavior is not defined. An example behavioral example describes the operational requirements for the host controller implementing park-mode. This feature does not affect how the host controller handles the bus transaction as defined in Execute Transaction . It only effects how many consecutive bus transactions for the current queue head can be executed. All boundary conditions, error detection and reporting applies as usual. This feature is similar in concept to the use of the *Mult* field for high-bandwidth Interrupt for queue heads in the Periodic Schedule.

The host controller effectively loads an internal down-counter *PM-Count* from *Asynchronous Schedule Park Mode Count* when *Asyncrhonous Schedule Park Mode Enable* bit is one, and a high-speed queue head is first fetched and meets all the criteria for executing a bus transaction. After the bus transaction, *PM-Count* is decremented. The host controller may continue to execute bus transactions from the current queue head until *PM-Count* goes to zero, an error is detected, the buffer for the current transfer is exhausted or the endpoint responds with a flow-control or STALL handshake.

Table 44-186 summarizes the responses that effect whether the host controller continues with another bus transaction for the current queue head.

Table 44-186. Actions for Park Mode, based on Endpoint Response and Residual Transfer State

PID	Endpoint Response	Transfer State after Transaction		Action
		PM-Count	Bytes to Transfer	

Table continues on the next page...

Table 44-186. Actions for Park Mode, based on Endpoint Response and Residual Transfer State (continued)

IN	DATA[0,1] w/Maximum Packet sized data	Not zero	Not Zero	Allowed to perform another bus transaction. [1],, [2]
		Not zero	Zero	Retire qTD and move to next QH
		Zero	Don't care	Move to next QH.
	DATA[0,1] w/short packet	Don't care	Don't care	Retire qTD and move to next QH.
	NAK	Don't care	Don't care	Move to next QH.
	STALL, XactErr	Don't care	Don't care	Move to next QH.
OUT	ACK	Not zero	Not Zero	Allowed to perform another bus transaction. [2]
		Not zero	Zero	Retire qTD and move to next QH
		Zero	Don't' care	Move to next QH.
	NYET, NAK	Don't care	Don't care	Move to next QH.
	STALL, XactErr	Don't care	Don't care	Move to next QH
PING	ACK	Not Zero	Not Zero	Allowed to perform another bus transaction. [2]
	NAK	Don't care	Don't care	Move to next QH
	STALL, XactErr	Don't care	Don't care	Move to next QH

1. The host controller may continue to execute bus transactions from the current high-speed queue head (if *PM-Count* is not equal to zero), if a PID mismatch is detected (for example expected DATA1 and received DATA0, or visa-versa).
2. This specification does not *require* that the host controller execute another bus transaction when *PM-Count* is non-zero. Implementations are encouraged to make appropriate complexity and performance trade-offs.

44.6.3.10.4 Write Back qTD

This state is entered from the Execute Transaction state when the *Active* bit is set to zero. The source data for the write-back is the transfer results area of the queue head overlay area (see Table 44-186). The host controller uses the *Current qTD Pointer* field as the target address for the qTD. The queue head transfer result area is written back to the transfer result area of the target qTD. This state is also referred to as: qTD retirement. The fields that must be written back to the source qTD include *Total Bytes to Transfer*, *Cerr*, and *Status*.

The duration of this state depends on when the qTD write-back is committed.

NOTE

If the retired qTD is the last one on the linked list (Terminate bit set on link pointers), then the host controller will read it one more time to check if the driver software updated the link pointers. If the memory holding the last qTD is re-used by software before the controller has read the qTD, the host controller may incorrectly use that data as a valid qTD and

perform incorrect actions or even crash. For this reason, driver software should not remove the last qTD from the linked list.

44.6.3.10.5 Follow Queue Head Horizontal Pointer

The host controller must use the horizontal pointer in the queue head to the next schedule data structure when any of the following conditions exist:

- If the *Active* bit is one on exit from the Execute Transaction state, or
- When the host controller exits the Write Back qTD state, or
- If the Advance Queue state fails to advance the queue because the target qTD is not active, or
- If the *Halted* bit is one on exit from the Fetch QH state.

There is no functional requirement that the host controller wait until the current transaction is complete before using the horizontal pointer to read the next linked data structure. However, it must wait until the current transaction is complete before executing the next data structure.

44.6.3.10.6 Buffer Pointer List Use for Data Streaming with qTDs

A qTD has an array of buffer pointers, which is used to reference the data buffer for a transfer. This specification requires that the buffer associated with the transfer be *virtually contiguous*. This means: if the buffer spans more than one physical page, it must obey the following rules (Figure 44-156 illustrates an example):

- The first portion of the buffer must begin at some offset in a page and extend through the end of the page.
- The remaining buffer cannot be allocated in small chunks scattered around memory. For each 4 K chunk beyond the first page, each buffer portion matches to a full 4 K page. The final portion, which may only be large enough to occupy a portion of a page, must start at the top of the page and be contiguous within that page.

The buffer pointer list in the qTD is long enough to support a maximum transfer size of 20 K bytes. This case occurs when all five buffer pointers are used and the first offset is zero. A qTD handles a 16 Kbyte buffer with any starting buffer alignment.

The host controller uses the field *C_Page* field as an index value to determine which buffer pointer in the list should be used to start the current transaction. The host controller uses a different buffer pointer for each physical page of the buffer. This is always true, even if the buffer is physically contiguous.

The host controller must detect when the current transaction spans a page boundary and automatically move to the next available buffer pointer in the page pointer list. The next available pointer is reached by incrementing *C_Page* and pulling the next page pointer from the list. Software must ensure there are sufficient buffer pointers to move the amount of data specified in the *Bytes to Transfer* field.

Figure 44-156 illustrates a nominal example of how System software would initialize the buffer pointers list and the *C_Page* field for a transfer size of 16383 bytes. *C_Page* is set to zero. The upper 20-bits of Page 0 references the start of the physical page. *Current Offset* (the lower 12-bits of queue head Dword 7) holds the offset in the page for example 2049 (for example 4096-2047). The remaining page pointers are set to reference the beginning of each subsequent 4 K page.

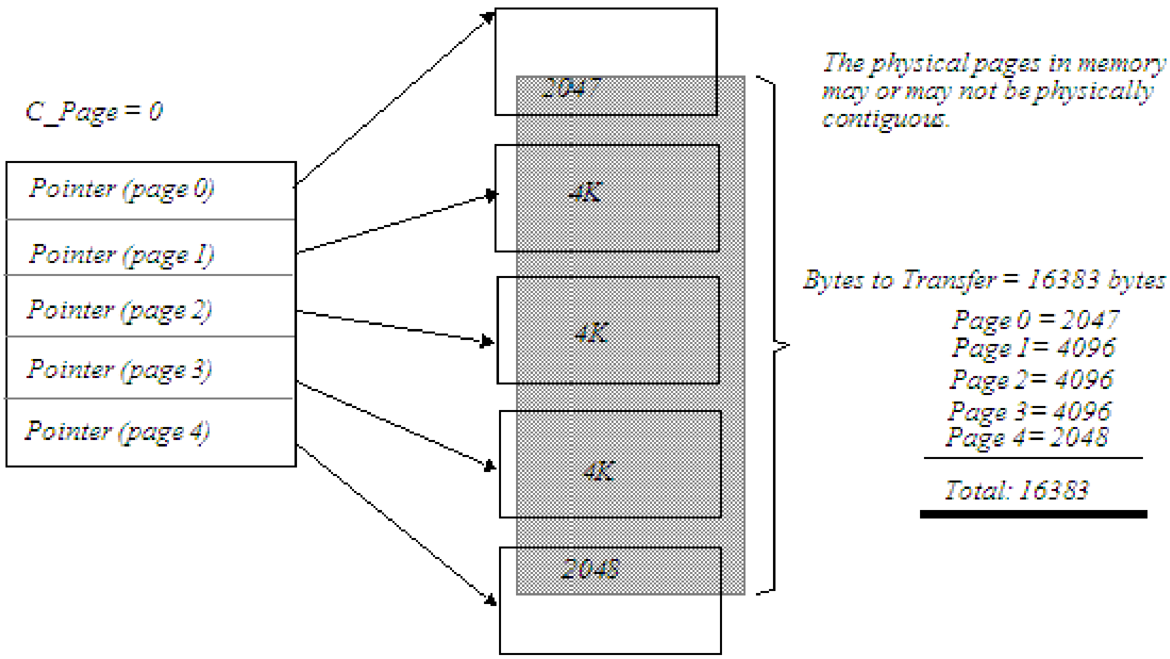


Figure 44-156. Example Mapping of qTD Buffer Pointers to Buffer Pages

For the first transaction on the qTD (assuming a 512-byte transaction), the host controller uses the first buffer pointer (page 0 because *C_Page* is set to zero) and concatenates the *Current Offset* field. The 512 bytes are moved during the transaction, the *Current Offset* and *Total Bytes to Transfer* are adjusted by 512 and written back to the queue head working area.

During the 4^{th} transaction, the host controller needs 511 bytes in page 0 and one byte in page 1. The host controller increments *C_Page* (to 1) and use the page 1 pointer to move the final byte of the transaction. After the 4^{th} transaction, the active page pointer is the page 1 pointer and *Current Offset* has rolled to one, and both are written back to the

overlay area. The transactions continue for the rest of the buffer, with the host controller automatically moving to the next page pointer (that is *C_Page*) when necessary. The three conditions for how the host controller handles *C_Page*:

- The current transaction does not span a page boundary. The value of *C_Page* is not adjusted by the host controller.
- The current transaction does span a page boundary. The host controller must detect the page cross condition and advance to the next buffer while streaming data to/from the USB.
- The current transaction completes on a page boundary (that is the last byte moved for the current transaction is the last byte in the page for the current page pointer). The host controller must increment *C_Page* before writing back status for the transaction.

NOTE

The only valid adjustment the host controller may make to *C_Page* is to increment by one.

44.6.3.10.7 Adding Interrupt Queue Heads to the Periodic Schedule

The link path(s) from the periodic frame list to a queue head establishes in which frames a transaction can be executed for the queue head. Queue heads are linked into the periodic schedule so they are polled at the appropriate rate. System software sets a bit in a queue head's *S-Mask* to indicate which micro-frame with-in 1 msec period a transaction should be executed for the queue head. Software must ensure that all queue heads in the periodic schedule have *S-Mask* set to a non-zero value. An *S-mask* with zero value in the context of the periodic schedule yields undefined results.

If the desired poll rate is greater than one frame, system software can use a combination of queue head linking and *S-Mask* values to spread interrupts of equal poll rates through the schedule so that the periodic bandwidth is allocated and managed in the most efficient manner possible. Some examples are illustrated in Table 44-187.

Table 44-187. Example Periodic Reference Patterns for Interrupt Transfers with 2ms Poll Rate

Frame # Reference Sequence	Description
0, 2, 4, 6, 8, and so on *S-Mask* = 01h	A queue head for the *bInterval* of 2 msec (16 micro-frames) is linked into the periodic schedule so that it is reachable from the periodic frame list locations indicated in the previous column. In addition, the *S-Mask* field in the queue head is set to 01h, indicating that the transaction for the endpoint should be executed on the bus during micro-frame 0 of the frame.
0, 2, 4, 6, 8, and so on *S-Mask* = 02h	Another example of a queue head with a *bInterval* of 2 msec is linked into the periodic frame list at exactly the same interval as the previous example. However, the *S-Mask* is set to 02h indicating that the transaction for the endpoint should be executed on the bus during micro-frame 1 of the frame.

44.6.3.10.8 Managing Transfer Complete Interrupts from Queue Heads

The host controller sets an interrupt to be signaled at the next interrupt threshold when the completed transfer (qTD) has an *Interrupt on Complete* (*IOC*) bit set to one, or whenever a transfer (qTD) completes with a short packet. If system software needs multiple qTDs to complete a client request (that is like a control transfer) the intermediate qTDs do not require interrupts. System software may only need a single interrupt to notify it that the complete buffer has been transferred. System software may set IOC's to occur more frequently. A motivation for this may be that it wants early notification so that interface data structures can be re-used in a timely manner.

44.6.3.11 Ping Control

USB 2.0 defines an addition to the protocol for high-speed devices called Ping. Ping is required for all USB 2.0 High-speed bulk and control endpoints. Ping is not allowed for a split-transaction stream. This extension to the protocol eliminates the bad side-effects of Naking OUT endpoints. The *Status* field has a *Ping State* bit, which the host controller uses to determine the *next* actual PID it uses in the next transaction to the endpoint (see Table 44-188).

The Ping State bit is only managed by the host controller for queue heads that meet the following criteria:

- Queue head is not an interrupt and
- *EPS* field equals High-Speed and
- *PIDCode* field equals OUT

Table 44-188 illustrates the state transition table for the host controller's responsibility for maintaining the PING protocol. Refer to Chapter 8 in the USB Specification Revision 2.0 for detailed description on the Ping protocol.

Table 44-188. Ping Control State Transition Table

Event			
Current	Host	Device	Next
Do Ping	PING	Nak	Do Ping
Do Ping	PING	Ack	Do OUT
Do Ping	PING	XactErr[1]	Do Ping
Do Ping	PING	Stall	N/C[2] Do
OUT	OUT	Nak	Do Ping
Do OUT	OUT	Nyet	Do Ping
Do OUT	OUT	Ack	Do OUT

Table continues on the next page...

Table 44-188. Ping Control State Transition Table (continued)

Do OUT	OUT	XactErr[1]	Do Ping
Do OUT	OUT	Stall	N/C[2]

1. Transaction Error (XactErr) is any time the host misses the handshake.
2. No transition change required for the Ping State bit. The Stall handshake results in the endpoint being halted (for example Active set to zero and Halt set to one). Software intervention is required to restart queue. 3 A Nyet response to an OUT means that the device has accepted the data, but cannot receive any more at this time. Host must advance the transfer state and additionally, transition the Ping State bit to Do Ping. The Ping State bit has the following encoding:

Table 44-189. Ping State bit Encoding

Value	Meaning
0B	Do OUT The host controller uses an OUT PID during the next bus transaction to this endpoint.
1B	Do Ping The host controller uses a PING PID during the next bus transaction to this endpoint.

The defined ping protocol (see USB 2.0 Specification, Chapter 8) allows the host to be *imprecise* on the initialization of the ping protocol (that is start in *Do OUT* when we don't know whether there is space on the device or not). The host controller manages the *Ping State* bit. System software sets the initial value in the queue head when it initializes a queue head. The host controller preserves the *Ping State* bit across all queue advancements. This means that when a new qTD is written into the queue head overlay area, the previous value of the *Ping State* bit is preserved.

44.6.3.12 Split Transactions

USB 2.0 defines extensions to the bus protocol for managing USB 1.x data streams through USB 2.0 Hubs. This section describes how the host controller uses the interface data structures to manage data streams with full- and low-speed devices, connected below USB 2.0 hub, utilizing the split transaction protocol. Refer to USB 2.0 Specification for the complete definition of the split transaction protocol. Full- and Low-speed devices are enumerated identically as high-speed devices, but the transactions to the Full- and Low-speed endpoints use the split-transaction protocol on the high-speed bus. The split transaction protocol is an encapsulation of (or wrapper around) the Full- or Low-speed transaction. The high-speed wrapper portion of the protocol is addressed to the USB 2.0 Hub and Transaction Translator below which the Full- or Low-speed device is attached.

The EHCI interface uses dedicated data structures for managing full-speed isochronous data streams (see Split Transaction Isochronous Transfer Descriptor (siTD)). Control, Bulk and Interrupt are managed using the queuing data structures (see Queue Head). The interface data structures need to be programmed with the device address and the

Transaction Translator number of the USB 2.0 Hub operating as the Low-/Full-speed host controller for this link. The following sections describe the details of how the host controller must process and manage the split transaction protocol.

44.6.3.12.1 Split Transactions for Asynchronous Transfers

A queue head in the asynchronous schedule with an *EPS* field indicating a full-or low-speed device indicates to the host controller that it must use split transactions to stream data for this queue head. All full-speed bulk and full-, low-speed control are managed through queue heads in the asynchronous schedule.

Software must initialize the queue head with the appropriate device address and port number for the transaction translator that is serving as the full/low-speed host controller for the links connecting the endpoint. Software must also initialize the split transaction state bit (*SplitXState*) to Do-Start-Split. Finally, if the endpoint is a control endpoint, then system software must set the *Control Transfer Type* (*C*) bit in the queue head to one. If this is not a control transfer type endpoint, the *C* bit must be initialized by software to be zero. This information is used by the host controller to properly set the Endpoint Type (ET) field in the split transaction bus token. When the *C* bit is zero, the split transaction token's ET field is set to indicate a bulk endpoint. When the *C* bit is one, the split transaction token's ET field is set to indicate a control endpoint. Refer to Chapter 8 of USB Specification Revision 2.0 for details.

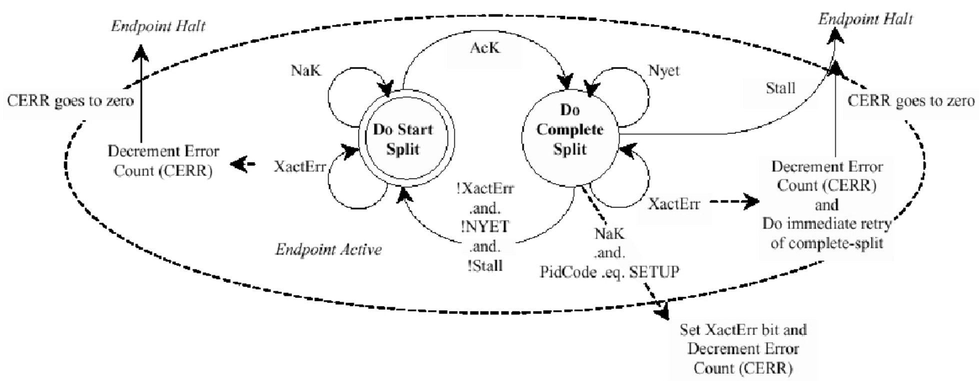


Figure 44-157. Host Controller Asynchronous Schedule Split-Transaction State Machine

44.6.3.12.1.1 Asynchronous - Do Start Split

This is the state which software must initialize a full- or low-speed asynchronous queue head. This state is entered from the Do Complete Split state only after a complete-split transaction receives a valid response from the transaction translator that is not a Nyet handshake.

For queue heads in this state, the host controller executes a start-split transaction to the appropriate transaction translator. If the bus transaction completes without an error and *PidCode* indicates an IN or OUT transaction, then the host controller reloads the error counter (*CErr*). If it is a successful bus transaction and the *PidCode* indicates a SETUP, the host controller does not reload the error counter. If the transaction translator responds with a Nak, the queue head is left in this state, and the host controller proceeds to the next queue head in the asynchronous schedule.

If the host controller times out the transaction (no response, or bad response) the host controller decrements *Cerr* and proceeds to the next queue head in the asynchronous schedule.

44.6.3.12.1.2 Asynchronous - Do Complete Split

This state is entered from the Do Start Split state only after a start-split transaction receives an Ack handshake from the transaction translator.

For queue heads in this state, the host controller executes a complete-split transaction to the appropriate transaction translator. If the transaction translator responds with a Nyet handshake, the queue head is left in this state, the error counter is reset and the host controller proceeds to the next queue head in the asynchronous schedule. When a Nyet handshake is received for a bus transaction where the queue head's *PidCode* indicates an IN or OUT, the host controller reloads the error counter (*CErr*). When a Nyet handshake is received for a complete-split bus transaction where the queue head's *PidCode* indicates a SETUP, the host controller must not adjust the value of *CErr*.

Independent of *PIDCode*, the following responses have the effects:

- Transaction Error (XactErr). Timeout or data CRC failure, and so on. The error counter (*Cerr*) is decremented by one and the complete split transaction is *immediately* retried (if possible). If there is not enough time in the micro-frame to execute the retry, the host controller MUST ensure that the next time the host controller begins executing from the Asynchronous schedule, it must begin executing from this queue head. If another start-split (for some other endpoint) is sent to the transaction translator before the complete-split is really completed, the transaction translator could dump the results (which were never delivered to the host). This is why the core specification states the retries must be immediate. A method to accomplish this behavior is to not advance the asynchronous schedule. When the host controller returns to the asynchronous schedule in the next micro-frame, the first transaction from the schedule is the retry for this endpoint.

If *Cerr* went to zero, the host controller must halt the queue.

- NAK. The target endpoint Nak'd the full- or low-speed transaction. The state of the transfer is not advanced and the state is exited. If the *PidCode* is a SETUP, then the Nak response is a protocol error. The *XactErr* status bit is set to one and the *CErr* field is decremented.
- STALL. The target endpoint responded with a STALL handshake. The host controller sets the *halt* bit in the status byte, retires the qTD but does not attempt to advance the queue.

If the *PidCode* indicates an IN, then any of following responses are expected:

- DATA0/1. On reception of data, the host controller ensures the PID matches the expected data toggle and checks CRC. If the packet is *good*, the host controller advances the state of the transfer, for example move the data pointer by the number of bytes received, decrement *BytesToTransfer* field by the number of bytes received, and toggle the *dt* bit. The host controller then exit this state. The response and advancement of transfer may trigger other processing events, such as retirement of the qTD and advancement of the queue.

If the data sequence PID does not match the expected, the data is ignored, the transfer state is not advanced and this state is exited. If the *PidCode* indicates an OUT/SETUP, then any of following responses are expected:

- ACK. The target endpoint accepted the data, so the host controller must advance the state of the transfer. The *Current Offset* field is incremented by *Maximum Packet Length* or *Bytes to Transfer*, whichever is less. The field *Bytes To Transfer* is decremented by the same amount and the data toggle bit (*dt*) is toggled. The host controller then exit this state.
- Advancing the transfer state may cause other processing events such as retirement of the qTD and advancement of the queue (see Managing Control/Bulk/Interrupt Transfers through Queue Heads).

44.6.3.12.2 Split Transaction Interrupt

Split-transaction Interrupt-IN/OUT endpoints are managed through the same data structures used for high-speed interrupt endpoints. They both co-exist in the periodic schedule. Queue heads/qTDs offer the set of features required for reliable data delivery, which is characteristic to interrupt transfer types. The split-transaction protocol is managed completely within this defined functional transfer framework. For example, for a high-speed endpoint, the host controller visits a queue head, execute a high-speed transaction (if criteria are met) and advance the transfer state (or not) depending on the results of the entire transaction. For low- and full-speed endpoints, the details of the *execution* phase are different (that is takes more than one bus transaction to complete),

but the remainder of the operational framework is intact. This means that the transfer advancement, and so on. occurs as defined in Managing Control/Bulk/Interrupt Transfers through Queue Heads, but only occurs on the completion of a split transaction.

44.6.3.12.2.1 Split Transaction Scheduling Mechanisms for Interrupt

Full- and low-speed Interrupt queue heads have an *EPS* field indicating full- or low-speed and have a non-zero *S-mask* field. The host controller can detect this combination of parameters and assume the endpoint is a periodic endpoint. Low- and full-speed interrupt queue heads require the use of the split transaction protocol. The host controller sets the Endpoint Type (ET) field in the split token to indicate the transaction is an interrupt. These transactions are managed through a transaction translator's periodic pipeline. Software should not set these fields to indicate the queue head is an interrupt unless the queue head is used in the periodic schedule.

System software manages the per/transaction translator periodic pipeline by budgeting and scheduling exactly during which micro-frames the start-splits and complete-splits for each endpoint occurs. The characteristics of the transaction translator are such that the high-speed transaction protocol must execute during explicit micro-frames, or the data or response information in the pipeline is lost.

Figure 44-158 illustrates the general scheduling boundary conditions that are supported by the EHCI periodic schedule and queue head data structure. The S and ^{C}X labels indicate micro-frames where software can schedule start-splits and complete splits (respectively).

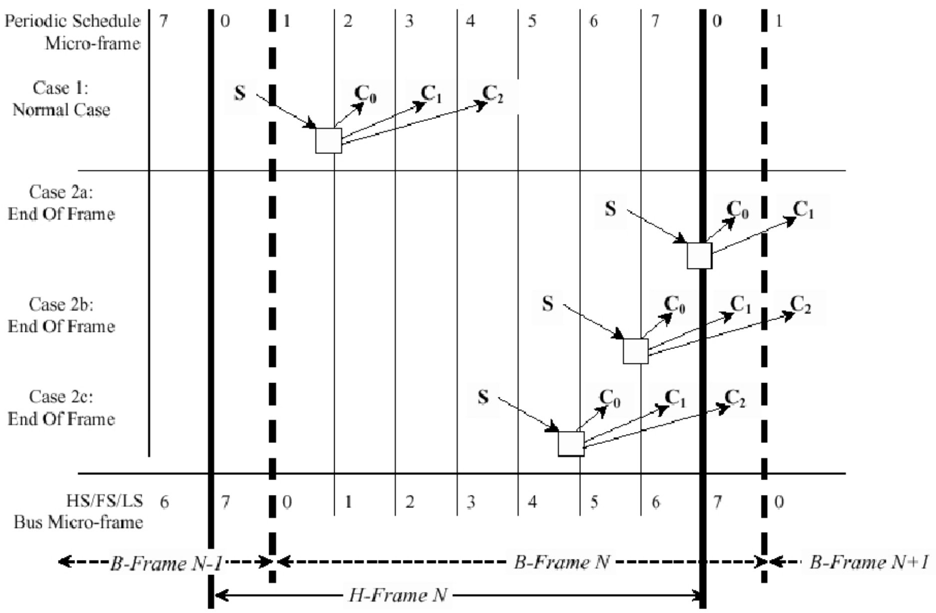


Figure 44-158. Split Transaction, Interrupt Scheduling Boundary Conditions

The scheduling cases are:

- Case 1: The normal scheduling case is where the entire split transaction is completely bounded by a frame (*H-Frame* in this case).
- Case 2a through Case 2c: The USB 2.0 Hub pipeline rules states clearly, when and how many complete-splits must be scheduled to account for earliest to latest execution on the full/low-speed link. The complete-splits may span the *H-Frame* boundary when the start-split is in micro-frame 4 or later. When this occurs, the *H-Frame* to *B-Frame* alignment requires that the queue head be reachable from consecutive periodic frame list locations. System software cannot build an efficient schedule that satisfies this requirement unless it uses FSTNs.

Figure 44-159 illustrates the general layout of the periodic schedule.

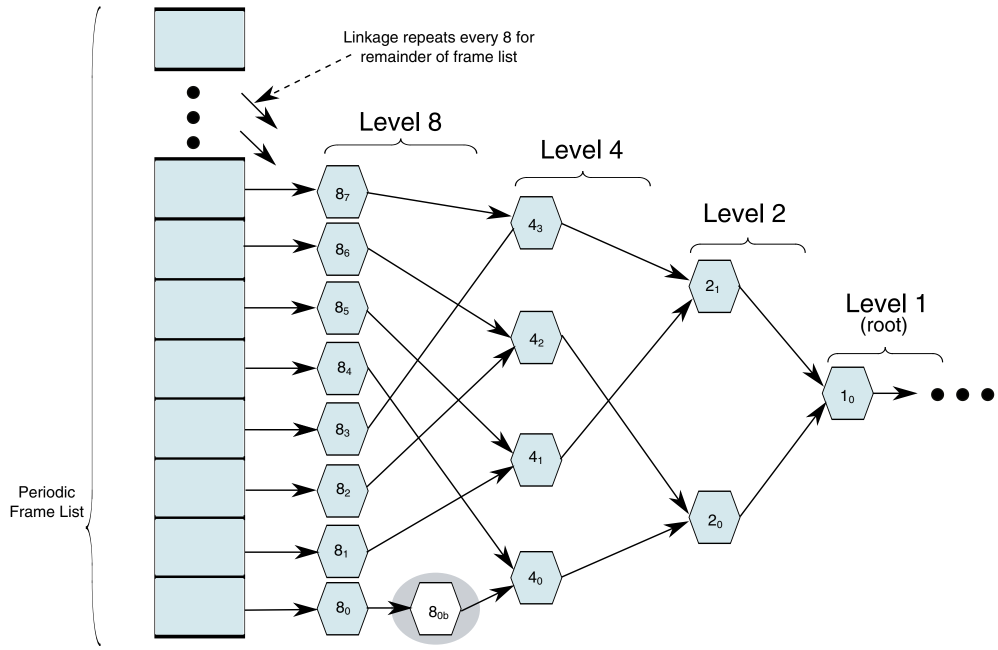


Figure 44-159. General Structure of EHCI Periodic Schedule Utilizing Interrupt Spreading

The periodic frame list is effectively the leaf level a binary tree, which is always traversed leaf to root. Each level in the tree corresponds to a 2^N poll rate. Software can efficiently manage periodic bandwidth on the USB by *spreading* interrupt queue heads that have the same poll rate requirement across all the available paths from the frame list. For example, system software can schedule eight poll rate 8 queue heads and account for them once in the high-speed bus bandwidth allocation.

When an endpoint is allocated an execution footprint that spans a frame boundary, the queue head for the endpoint must be reachable from consecutive locations in the frame list. An example would be if 8_{0b} where such an endpoint. Without additional support on the interface, to get 8_{0b} reachable at the correct time, software would have to link 8_1 to

8_{0b}. It would then have to move 4_1 and everything linked after into the same path as 4_0. This upsets the integrity of the binary tree and disallows the use of the spreading technique.

FSTN data structures are used to preserve the integrity of the binary-tree structure and enable the use of the spreading technique. Host Controller Operational Model for FSTNs defines the hardware and software operational model requirements for using FSTNs.

The following queue head fields are initialized by system software to instruct the host controller when to execute portions of the split-transaction protocol:

- *SplitXState*. This is single bit residing in the *Status* field of a queue head (see Table 44-166). This bit is used to track the current state of the split transaction.
- *Frame S-mask*. This is a field where-in system software sets a bit corresponding to the micro-frame (within an *H-Frame*) that the host controller should execute a start-split transaction. This is always qualified by the value of the *SplitXState* bit in the *Status* field of the queue head. For example, referring to Figure 44-158, case one, the *S-mask* would have a value of 00000001b indicating that if the queue head is traversed by the host controller, and the *SplitXState* indicates Do_Start, and the current micro-frame as indicated by FRINDEX[2:0] is 0, then execute a start-split transaction.
- *Frame C-mask.* This is a field where system software sets one or more bits corresponding to the micro-frames (within an *H-Frame*) that the host controller should execute complete-split transactions. The interpretation of this field is always qualified by the value of the *SplitXState* bit in the *Status* field of the queue head. For example, referring to Figure 44-158, case one, the *C-mask* would have a value of 00011100b indicating that if the queue head is traversed by the host controller, and the *SplitXState* indicates Do_Complete, and the current micro-frame as indicated by FRINDEX[2:0] is 2, 3, or 4, then execute a complete-split transaction. It is software's responsibility to ensure that the translation between *H-Frames* and *B-Frames* is correctly performed when setting bits in *S-mask* and *C-mask*

44.6.3.12.2.2 Host Controller Operational Model for FSTNs

The FSTN data structure is used to manage Low/Full-speed interrupt queue heads that need to be reached from consecutive frame list locations (that is boundary cases 2a through 2c). An FSTN is essentially a *back pointer*, similar in intent to the back pointer field in the siTD data structure (see siTD Back Link Pointer).

This feature provides software a simple primitive to save a schedule position, redirect the host controller to traverse the necessary queue heads in the previous frame, then restore the original schedule position and complete normal traversal.

The four components to the use of FSTNs:

- FSTN data structure.
- A *Save Place* indicator. This is always an FSTN with its *Back Path Link Pointer.T-bit* set to zero.
- A *Restore* indicator. This is always an FSTN with its *Back Path Link Pointer.T-bit* set to one.
- Host controller FSTN traversal rules.

When the host controller encounters an FSTN during micro-frames 2 through 7 it simply follows the node's *Normal Path Link Pointer* to access the next schedule data structure.

NOTE

The FSTN's *Normal Path Link Pointer.T-bit* may set to one, which the host controller must interpret as the end of periodic list mark.

When the host controller encounters a *Save-Place* FSTN in micro-frames 0 or 1, it saves the value of the *Normal Path Link Pointer* and set an internal flag indicating that it is executing in *Recovery Path* mode. *Recovery Path* mode modifies the host controller's rules for how it traverses the schedule and limits which data structures is considered for execution of bus transactions. The host controller continues executing in *Recovery Path* mode until it encounters a *Restore* FSTN or it determines that it has reached the end of the micro-frame (see details in the list below).

The rules for schedule traversal and limited execution while in *Recovery Path* mode are:

- Always follow the *Normal Path Link Pointer* when it encounters an FSTN that is a *Save-Place* indicator. The host controller must not recursively follow *Save-Place* FSTNs. Therefore, while executing in *Recovery Path* mode, it must never follow an FSTN's *Back Path Link Pointer*.
- Do not process an siTD or, iTD data structure. Simply follow its *Next Link Pointer*.
- Do not process a QH (Queue Head) whose *EPS* field indicates a high-speed device. Simply follow its *Horizontal Link Pointer*.
- When a QH's *EPS* field indicates a Full/Low-speed device, the host controller considers only it for execution if its *SplitXState* is DoComplete (note: this applies whether the *PID Code* indicates an IN or an OUT). See Execute Transaction and Tracking Split Transaction Progress for Interrupt Transfers for a complete list of additional conditions that must be met in general for the host controller to issue a bus transaction.
    - The host controller must not execute a Start-split transaction while executing in *Recovery Path* mode. See Periodic Isochronous - Do Complete Split for special handling when in *Recovery Path* mode.
- Stop traversing the *recovery path* when it encounters an FSTN that is a *Restore* indicator. The host controller unconditionally uses the saved value of the *Save-Place* FSTN's *Normal Path Link Pointer* when returning to the normal path traversal. The

host controller must clear the context of executing a *Recovery Path* when it restores schedule traversal to the *Save-Place* FSTN's *Normal Path Link Pointer*.

- If the host controller determines that there is not enough time left in the micro-frame to complete processing of the periodic schedule, it abandons traversal of the recovery path, and clears the context of executing a recovery path. The result is that at the start of the next consecutive micro-frame, the host controller starts traversal at the frame list.

An example traversal of a periodic schedule that includes FSTNs is illustrated in Figure 44-160.

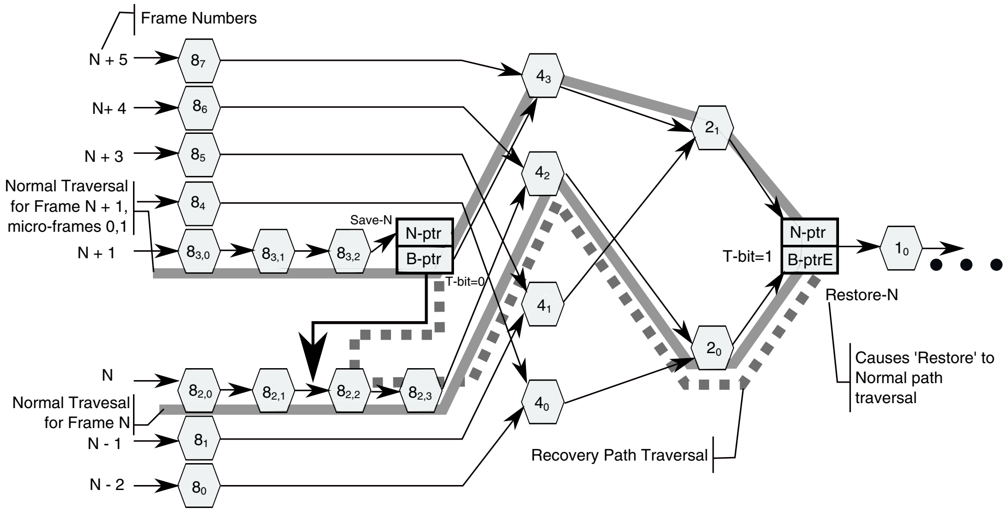


Figure 44-160. Example Host Controller Traversal of Recovery Path via FSTNs

In frame N+1 (micro-frames 0 and 1), when the host controller encounters Save-Path FSTN (Save-N), it observes that Save-N.Back Path Link Pointer.T-bit is zero (definition of a Save-Path indicator). The host controller saves the value of Save-N.Normal Path Link Pointer and follows Save-N.Back Path Link Pointer. At the same time, it sets an internal flag indicating that it is now in *Recovery Path* mode (the recovery path is annotated in Figure 44-160 with a large dashed line). The host controller continues traversing data structures on the recovery path and executing only those bus transactions as noted above, on the recovery path until it reaches Restore FSTN (Restore-N). Restore-N.Back Path Link Pointer.T-bit is set to one (definition of a Restore indicator), so the host controller exits *Recovery Path* mode by clearing the internal *Recovery Path* mode flag and commences (restores) schedule traversal using the saved value of the *Save-Place* FSTN's *Normal Path Link Pointer* (for example Save-N.Normal Path Link Pointer). The nodes traversed during these micro-frames include: {$8_{3.0}$, $8_{3.1}$, $8_{3.2}$, Save-A, $8_{2.2}$, $8_{2.3}$, 4_2, 2_0, Restore-N, 4_3, 2_1, Restore-N, 1_0 }. The nodes on the recovery-path are bolded. In frame N (micro-frames 0-7), for this example, the host controller traverses all of the schedule data structures utilizing the *Normal Path Link Pointers* in any FSTNs it encounters. This is because the host controller has not yet encountered a *Save-Place*

FSTN so it not executing in *Recovery Path* mode. When it encounters the *Restore* FSTN, (Restore-N), during micro-frames 0 and 1, it uses Restore-N.Normal Path Link Pointer to traverse to the next data structure (that is normal schedule traversal). This is because the host controller must use a Restore FSTN's *Normal Path Link Pointer* when not executing in a *Recovery-Path* mode. The nodes traversed during frame N include: {$8_{2.0}$, $8_{2.1}$, $8_{2.2}$, $8_{2.3}$, 4_2, 2_0, Restore-N, 1_0 }.

In frame N+1 (micro-frames 2-7), when the host controller encounters Save-Path FSTN Save-N, it unconditionally follows Save-N.Normal Path Link Pointer. The nodes traversed during these micro-frames include: {$8_{3.0}$, $8_{3.1}$, $8_{3.2}$, Save-A, 4_3, 2_1, Restore-N, 1_0 }.

44.6.3.12.2.3 Software Operational Model for FSTNs

Software must create a consistent, coherent schedule for the host controller to traverse. When using FSTNs, system software must adhere to the following rules:

- Each *Save-Place* indicator requires a matching *Restore* indicator.
  - The *Save-Place* indicator is an FSTN with a valid *Back Path Link Pointer and T-bit* equal to zero.
    - *Back Path Link Pointer.Typ* field must be set to indicate the referenced data structure is a queue head. The *Restore* indicator is an FSTN with its *Back Path Link Pointer.T-bit* set to one.
  - A *Restore* FSTN may be matched to one or more *Save-Place* FSTNs. For example, if the schedule includes a poll-rate 1 level, then system software only needs to place a *Restore* FSTN at the beginning of this list in order to match all possible *Save-Place* FSTNs.
- If the schedule does not have elements linked at a poll-rate level of one, and one or more *Save-Place* FSTNs are used, then System Software must ensure the *Restore* FSTN's *Normal Path Link Pointer*'s *T-bit* is set to one, as this is used to mark the end of the periodic list.
- When the schedule does have elements linked at a poll rate level of one, a *Restore* FSTN must be the first data structure on the poll rate one list. All traversal paths from the frame list converge on the poll-rate one list. System software must ensure that *Recovery Path* mode is exited before the host controller is allowed to traverse the poll rate level one list.
- A *Save-Place* FSTN's *Back Path Link Pointer* must reference a queue head data structure. The referenced queue head must be reachable from the previous frame list location. In other words, if the *Save-Place* FSTN is reachable from frame list offset N, then the FSTN's *Back Path Link Pointer* must reference a queue head that is reachable from frame list offset N-1.

Software should make the schedule as efficient as possible. What this means in this context is that software should have no more than one *Save-Place* FSTN reachable in any single frame. Note there is times when two (or more, depending on the implementation) could exist as full/low-speed footprints change with bandwidth adjustments. This could occur, for example when a bandwidth rebalance causes system software to move the *Save-Place* FSTN from one poll rate level to another. During the transition, software must preserve the integrity of the previous schedule until the new schedule is in place.

44.6.3.12.2.4 Tracking Split Transaction Progress for Interrupt Transfers

To correctly maintain the data stream, the host controller must be able to detect and report errors where data is lost. For interrupt-IN transfers, data is lost when it makes it into the USB 2.0 hub, but the USB 2.0 host system is unable to get it from the USB 2.0 Hub and into the system before it expires from the transaction translator pipeline. When a lost data condition is detected, the queue must be halted, thus signaling system software to recover from the error. A data-loss condition exists whenever a start-split is issued, accepted and successfully executed by the USB 2.0 Hub, but the complete-splits get unrecoverable errors on the high-speed link, or the complete-splits do not occur at the correct times. One reason complete-splits might not occur at the right time would be due to host-induced system hold-offs that cause the host controller to miss bus transactions because it cannot get timely access to the schedule in system memory.

The same condition can occur for an interrupt-OUT, but the result is not an endpoint halt condition, but rather effects only the progress of the transfer. The queue head has the following fields to track the progress of each split transaction. These fields are used to keep incremental state about which (and when) portions have been executed.

- *C-prog-mask*. This is an eight-bit bit-vector where the host controller keeps track of which complete-splits have been executed. Due to the nature of the Transaction Translator periodic pipeline, the complete-splits need to be executed in-order. The host controller needs to detect when the complete-splits have not been executed in order. This can only occur due to system hold-offs where the host controller cannot get to the memory-based schedule. *C-prog-mask* is a simple bit-vector that the host controller sets one of the *C-prog*-mask bits for each complete-split executed. The bit position is determined by the micro-frame number in which the complete-split was executed. The host controller always checks *C-prog-mask* before executing a complete-split transaction. If the previous complete-splits have not been executed then it means one (or more) have been skipped and data has potentially been lost.

- *FrameTag*. This field is used by the host controller during the complete-split portion of the split transaction to tag the queue head with the frame number (*H-Frame* number) when the next complete split must be executed.
- *S-bytes*. This field can be used to store the number of data payload bytes sent during the start-split (if the transaction was an OUT). The *S-bytes* field must be used to accumulate the data payload bytes received during the complete-splits (for an IN).

44.6.3.12.2.5 Split Transaction Execution State Machine for Interrupt

In the following presentation, all references to micro-frame are in the context of a micro-frame within an *H-Frame.*

As with asynchronous Full- and Low-speed endpoints, a split-transaction state machine is used to manage the split transaction sequence. Aside from the fields defined in the queue head for scheduling and tracking the split transaction, the host controller calculates one internal mechanism that is also used to manage the split transaction. The internal calculated mechanism is:

- *cMicroFrameBit*. This is a single-bit encoding of the current micro-frame number. It is an eight-bit value calculated by the host controller at the beginning of every micro-frame. It is calculated from the three least significant bits of the *FRINDEX* register (that is, *cMicroFrameBit* = (1 shifted-left(*FRINDEX*[2:0])). The *cMicroFrameBit* has at most one bit asserted, which always corresponds to the current micro-frame number. For example, if the current micro-frame is 0, then *cMicroFrameBit* will equal 00000001b. The variable *cMicroFrameBit* is used to compare against the *S-mask* and *C-mask* fields to determine whether the queue head is marked for a start- or complete-splt transaction for the current micro-frame.

Figure 44-161 illustrates the state machine for managing a complete interrupt split transaction. There are two phases to each split transaction. The first is a single start-split transaction, which occurs when the *SplitXState* is at Do_Start and the single bit in *cMicroFrameBit* has a corresponding bit active in *QH.S-mask*. The transaction translator does not acknowledge the receipt of the periodic start-split, so the host controller unconditionally transitions the state to Do_Complete. Due to the available jitter in the transaction translator pipeline, there will be more than one complete-split transaction scheduled by software for the Do_Complete state. This translates simply to the fact that there are multiple bits set to a one in the *QH.C-mask* field.

The host controller keeps the queue head in the Do_Complete state until the split transaction is complete (see definition below), or an error condition triggers the *three-strikes-rule* (for example, after the host tries the same transaction three times, and each

encounters an error, the host controller will stop retrying the bus transaction and halt the endpoint, thus requiring system software to detect the condition and perform system-dependent recovery).

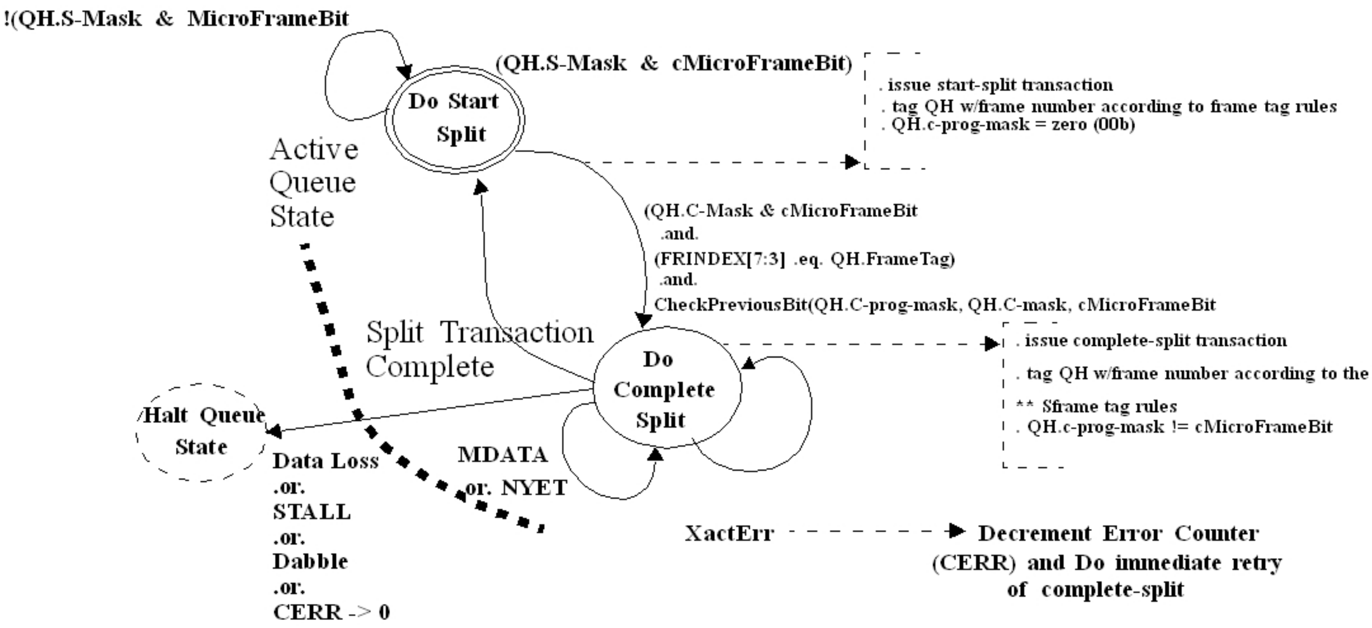


Figure 44-161. Split Transaction State Machine for Interrupt

See Previous Section for the frame tag management rules.

Periodic Interrupt - Do Start Split

This is the state software must initialize a full- or low-speed interrupt queue head *StartXState* bit. This state is entered from the Do_Complete Split state only after the split transaction is complete. This occurs when one of the following events occur: The transaction translator responds to a complete-split transaction with one of the following:

- NAK. A NAK response is a propagation of the full- or low-speed endpoint's NAK response.
- ACK. An ACK response is a propagation of the full- or low-speed endpoint's ACK response. Only occurs on an OUT endpoint.
- DATA 0/1. Only occurs for INs. Indicates that this is the last of the data from the endpoint for this split transaction.
- ERR. The transaction on the low-/full-speed link below the transaction translator had a failure (for example, timeout, bad CRC, etc.).
- NYET (and Last). The host controller issued the last complete-split and the transaction translator responded with a NYET handshake. This means that the start-split was not correctly received by the transaction translator, so it never executed a transaction to the full- or low-speed endpoint, see Section Periodic Isochronous - Do Complete Split for the definition of 'Last'.

Each time the host controller visits a queue head in this state (once within the Execute Transaction state), it performs the following test to determine whether to execute a start-split.

- *QH.S-mask* is bit-wise anded with *cMicroFrameBit*.

If the result is non-zero, then the host controller will issue a start-split transaction. If the *PIDCode* field indicates an IN transaction, the host controller must zero-out the *QH.S-bytes* field. After the split-transaction has been executed, the host controller sets up state in the queue head to track the progress of the complete-split phase of the split transaction. Specifically, it records the expected frame number into *QH.FrameTag* field (see Section), set *C-prog-mask* to zero (00h), and exits this state. Note that the host controller must not adjust the value of *CErr* as a result of completion of a start-split transaction.

Periodic Interrupt - Do Complete Split

This state is entered unconditionally from the Do Start Split state after a start-split transaction is executed on the bus. Each time the host controller visits a queue head in this state (once within the Execute Transaction state), it checks to determine whether a complete-split transaction should be executed now.

There are four tests to determine whether a complete-split transaction should be executed.

- Test A. *cMicroFrameBit* is bit-wise anded with *QH.C-mask* field. A non-zero result indicates that software scheduled a complete-split for this endpoint, during this micro-frame.
- Test B. *QH.FrameTag* is compared with the current contents of *FRINDEX*[7:3]. An equal indicates a match.
- Test C. The complete-split progress bit vector is checked to determine whether the previous bit is set, indicating that the previous complete-split was appropriately executed. An example algorithm for this test is provided below:

```
Algorithm Boolean CheckPreviousBit(QH.C-prog-mask, QH.C-mask, cMicroFrameBit)
Begin
-- Return values:
-- TRUE - no error
-- FALSE - error
--
Boolean rvalue = TRUE;
previousBit = cMicroframeBit logical-rotate-right(1)
-- Bit-wise anding previousBit with C-mask indicates
-- whether there was an intent
-- to send a complete split in the previous micro-frame. So,
-- if the
-- 'previous bit' is set in C-mask, check C-prog-mask to
-- make sure it
-- happened.
If (previousBit bitAND QH.C-mask)then
                               If not(previousBit bitAND QH.C-prog-mask) then
                    rvalue = FALSE;
          End if
End If
-- If the C-prog-mask already has a one in this bit position,
```

```
-- then an aliasing
-- error has occurred. It will probably get caught by the
-- FrameTag Test, but
-- at any rate it is an error condition that as detectable here
-- should not allow
-- a transaction to be executed.
If (cMicroFrameBit bitAND QH.C-prog-mask) then
 rvalue = FALSE;
End if
return (rvalue)
End Algorithm
```

- Test D. Check to see if a start-split should be executed in this micro-frame. Note this is the same test performed in the Do Start Split state (see Section Periodic Isochronous - Do Start Split). Whenever it evaluates to TRUE and the controller is NOT processing in the context of a *Recovery Path* mode, it means a start-split should occur in this micro-frame. Test D and Test A evaluating to TRUE at the same time is a system software error. Behavior is undefined.

If (A .and. B .and. C .and. not(D)) then the host controller will execute a complete-split transaction. When the host controller commits to executing the complete-split transaction, it updates *QH.C-prog-mask* by bit-ORing with *cMicroFrameBit*. On completion of the complete-split transaction, the host controller records the result of the transaction in the queue head and sets *QH.FrameTag* to the expected *H-Frame* number (see Section). The effect to the state of the queue head and thus the state of the transfer depends on the response by the transaction translator to the complete-split transaction. The following responses have the effects (note that any responses that result in decrementing of the *CErr* will result in the queue head being halted by the host controller if the result of the decrement is zero):

- NYET (and Last). On each NYET response, the host controller checks to determine whether this is the last complete-split for this split transaction. Last is defined in this context as the condition where all of the scheduled complete-splits have been executed. If it is the last complete-split (with a NYET response), then the transfer state of the queue head is not advanced (never received any data) and this state exited. The transaction translator must have responded to all the clompete-splits with NYETs, meaning that the start-split issued by the host controller was not received. The start-split should be retried at the next poll period.
- The test for whether this is the Last complete split can be performed by XOR *QH.C-mask* with *QH.C-prog-mask*. If the result is all zeros then all complete-splits have been executed. When this condition occurs, the *XactErr* status bit is set to a one and the *CErr* field is decremented.
- NYET (and not Last). See above description for testing for Last. The complete-split transaction received a NYET response from the transaction translator. Do not update any transfer state (except for *C-prog-mask* and *FrameTag*) and stay in this state. The host controller must not adjust *CErr* on this response.

- Transaction Error (XactErr). Timeout, data CRC failure, etc. The *CErr* field is decremented and the *XactErr* bit in the *Status* field is set to a one. The complete split transaction is *immediately* retried (if *Cerr* is non-zero).If there is not enough time in the micro-frame to complete the retry and the endpoint is an IN, or *CErr* is decremented to a zero from a one, the queue is halted. If there is not enough time in the micro-frame to complete the retry and the endpoint is an OUT and *CErr* is not zero, then this state is exited (that is, return to Do Start Split). This results in a retry of the entire OUT split transaction, at the next poll period. Refer to Chapter 11 Hubs (specifically the section full- and low-speed Interrupts) in the USB Specification Revision 2.0 for detailed requirements on why these errors must be immediately retried.
- ACK. This can only occur if the target endpoint is an OUT. The target endpoint ACK'd the data and this response is a propagation of the endpoint ACK up to the host controller. The host controller must advance the state of the transfer. The *Current Offset* field is incremented by *Maximum Packet Length* or *Bytes to Transfer*, whichever is less. The field *Bytes To Transfer* is decremented by the same amount. And the data toggle bit (*dt*) is toggled. The host controller will then exit this state for this queue head. The host controller must reload *CErr* with maximum value on this response. Advancing the transfer state may cause other process events such as retirement of the qTD and advancement of the queue (see Section Managing Control/Bulk/Interrupt Transfers through Queue Heads).
- MDATA. This response will only occur for an IN endpoint. The transaction translator responded with zero or more bytes of data and an MDATA PID. The incremental number of bytes received is accumulated in *QH.S-bytes*. The host controller must not adjust *CErr* on this response.
- DATA0/1. This response may only occur for an IN endpoint. The number of bytes received is added to the accumulated byte count in *QH.S-bytes*. The state of the transfer is advanced by the result and the host controller will exit this state for this queue head.
- Advancing the transfer state may cause other processing events such as retirement of the qTD and advancement of the queue (see Section Managing Control/Bulk/Interrupt Transfers through Queue Heads).
- If the data sequence PID does not match the expected, the entirety of the data received in this split transaction is ignored, the transfer state is not advanced and this state is exited.
- NAK. The target endpoint Nak'd the full- or low-speed transaction. The state of the transfer is not advanced, and this state is exited. The host controller must reload *CErr* with maximum value on this response.

- ERR. There was an error during the full- or low-speed transaction. The ERR status bit is set to a one, *Cerr* is decremented, the state of the transfer is not advanced, and this state is exited.
- STALL. The queue is halted (an exit condition of the Execute Transaction state). The status field bits: *Active* bit is set to zero and the *Halted* bit is set to a one and the qTD is retired. Responses which are not enumerated in the list or which are received out of sequence are illegal and may result in undefined host controller behavior. The other possible combinations of tests A, B, C, and D may indicate that data or response was lost. Table 44-190 lists the possible combinations and the appropriate action.

Table 44-190. Interrupt IN/OUT Do Complete Split State Execution Criteria

Condition	Action	Description
not(A) not(D)	Ignore QHD	Neither a start nor complete-split is scheduled for the current micro-frame.Host controller should continue walking the schedule.
A not(C)	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	Progress bit check failed. These means a complete-split has been missed. There is the possibility of lost data. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one.
A not(B) C	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	*QH.FrameTag* test failed. This means that exactly one or more *H-Frames* have been skipped. This means complete-splits and have missed. There is the possibility of lost data. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one.
A B C not(D)	Execute complete-split	This is the non-error case where the host controller executes a complete-split transaction.

Table continues on the next page...

Table 44-190. Interrupt IN/OUT Do Complete Split State Execution Criteria (continued)

D	If PIDCode = IN Halt QHD If PIDCode = OUT Retry start-split	This is a degenerate case where the start-split was issued, but all of the complete-splits were skipped and all possible intervening opportunities to detect the missed data failed to fire. If *PIDCode* is an IN, then the Queue head must be halted. If *PIDCode* is an OUT, then the transfer state is not advanced and the state exited (for example, start-split is retried). This is a host-induced error and does not effect *CER*R. In either case, set the *Missed Micro-frame* bit in the status field to a one. Note: When executing in the context of a *Recovery Path* mode, the host controller is allowed to process the queue head and take the actions indicated above, or it may wait until the queue head is visited in the normal processing mode. Regardless, the host controller must not execute a start-split in the context of a executing in a *Recovery Path* mode.

Managing QH.FrameTag Field

The *QH.FrameTag* field in a queue head is completely managed by the host controller. The rules for setting *QH.FrameTag* are simple:

- Rule 1: If transitioning from Do Start Split to Do Complete Split and the current value of *FRINDEX*[2:0] is 6 *QH.FrameTag* is set to *FRINDEX*[7:3] + 1. This accommodates split transactions whose start-split and complete-splits are in different *H-Frames* (case 2a, see Figure 44-158).
- Rule 2: If the current value of *FRINDEX*[2:0] is 7, *QH.FrameTag* is set to *FRINDEX*[7:3] + 1. This accommodates staying in Do Complete Split for cases 2a, 2b, and 2c (Figure 44-158).
- Rule 3: If transitioning from Do_Start Split to Do Complete Split and the current value of *FRINDEX*[2:0] is not 6, or currently in Do Complete Split and the current value of (*FRINDEX*[2:0]) is not 7, *FrameTag* is set to *FRINDEX*[7:3]. This accommodates all other cases (Figure 44-158).

44.6.3.12.2.6 Rebalancing the Periodic Schedule

System software must occasionally adjust a periodic queue head's S-mask and C-mask fields during operation. This need occurs when adjustments to the periodic schedule create a new bandwidth budget and one or more queue head's are assigned new execution footprints (that is, new S-mask and C-mask values).

It is imperative that System software must not update these masks to new values in the midst of a split transaction. In order to avoid any race conditions with the update, the EHCI host controller provides a simple assist to system software. System software sets the *Inactivate-on-next-Transaction* (*I*) bit to a one to signal the host controller that it intends to update the S-mask and C-mask on this queue head. System software will then

wait for the host controller to observe the *I-bit* is a one and transition the *Active* bit to a zero. The rules for how and when the host controller sets the *Active* bit to zero are enumerated below:

- If the *Active* bit is a zero, no action is taken. The host controller does not attempt to advance the queue when the *I-bit* is a one.
- If the *Active* bit is a one and the *SplitXState* is DoStart (regardless of the value of *S-mask*), the host controller will simply set *Active* bit to a zero. The host controller is not required to write the transfer state back to the *current* qTD. Note that if the *S-mask* indicates that a start-split is scheduled for the current micro-frame, the host controller must not issue the start-split bus transaction. It must set the *Active* bit to zero.

System software must save transfer state before setting the *I-bit* to a one. This is required so that it can correctly determine what transfer progress (if any) occurred after the *I-bit* was set to a one and the host controller executed its final bus-transaction and set *Active* to a zero.

After system software has updated the S-mask and C-mask, it must then reactivate the queue head. Because the *Active* bit and the *I-bit* cannot be updated with the same write, system software needs to use the following algorithm to coherently re-activate a queue head that has been stopped via the *I-bit*.

1. Set the *Halted* bit to a one, then
2. Set the *I-bit* to a zero, then
3. Set the *Active* bit to a one and the *Halted* bit to a zero in the same write.

Setting the *Halted* bit to a one inhibits the host controller from attempting to advance the queue between the time the *I-bit* goes to a zero and the *Active* bit goes to a one.

44.6.3.12.3 Split Transaction Isochronous

Full-speed isochronous transfers are managed using the split-transaction protocol through a USB 2.0 transaction translator in a USB2.0 Hub. The EHCI controller utilizes siTD data structure to support the special requirements of isochronous split-transactions. This data structure uses the scheduling model of isochronous TDs (iTD, Section Isochronous (High-Speed) Transfer Descriptor (iTD)) (see Section Managing Isochronous Transfers Using iTDs for the operational model of iTDs) with the contiguous data feature provided by queue heads. This simple arrangement allows a single isochronous scheduling model and adds the additional feature that all data received from the endpoint (per split transaction) must land into a contiguous buffer.

44.6.3.12.3.1 Split Transaction Scheduling Mechanisms for Isochronous

Full-speed isochronous transactions are managed through a transaction translator's periodic pipeline. As with full- and low-speed interrupt, system software manages each transaction translator's periodic pipeline by budgeting and scheduling exactly during which micro-frames the start-splits and complete-splits for each full-speed isochronous endpoint occur. The requirements described in Section Split Transaction Scheduling Mechanisms for Interrupt apply. Figure 44-162 illustrates the general scheduling boundary conditions that are supported by the EHCI periodic schedule. The ^{S}X and ^{C}X labels indicate micro-frames where software can schedule start- and complete-splits (respectively). The *H-Frame* boundaries are marked with a large, solid bold vertical line. The *B-Frame* boundaries are marked with a large, bold, dashed line. The bottom of the figure illustrates the relationship of an siTD to the *H-Frame*.

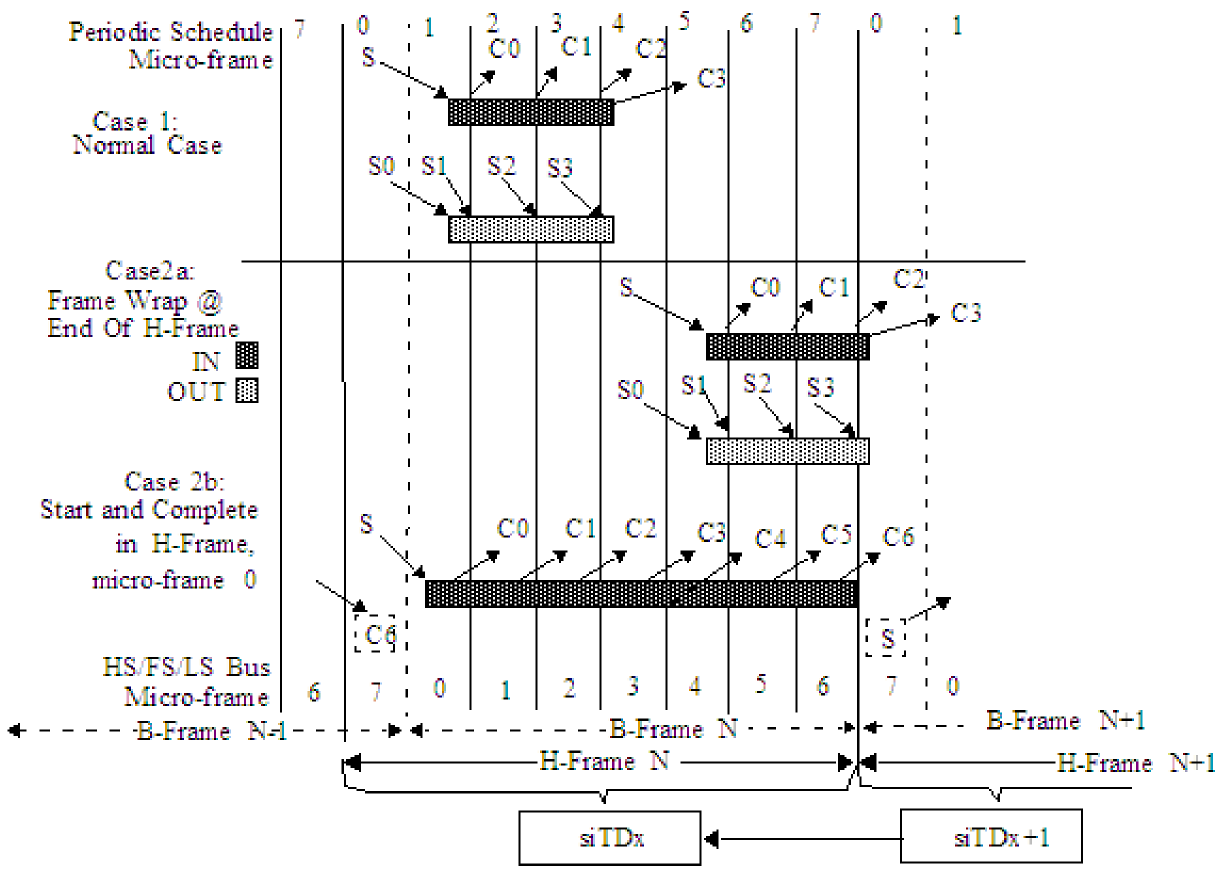


Figure 44-162. Split Transaction, Isochronous Scheduling Boundary Conditions

When the endpoint is an isochronous OUT, there are only start-splits, and no complete-splits. When the endpoint is an isochronous IN, there is at most one start-split and one to *N* complete-splits. The scheduling boundary cases are:

- *Case 1*: The entire split transaction is completely bounded by an *H-Frame*. For example: the start-splits and complete-splits are all scheduled to occur in the same *H-Frame*.
- *Case 2a*: This boundary case is where one or more (at most two) complete-splits of a split transaction IN are scheduled across an *H-Frame* boundary. This can only occur when the split transaction has the possibility of moving data in *B-Frame,* micro-frames 6 or 7 (*H-Frame* micro-frame 7 or 0). When an *H-Frame* boundary wrap condition occurs, the scheduling of the split transaction spans more than one location in the periodic list.(For example, it takes two siTDs in adjacent periodic frame list locations to fully describe the scheduling for the split transaction.)
- Although the scheduling of the split transaction may take two data structures, all of the complete-splits for each full-speed IN isochronous transaction must use only one data pointer. For this reason, siTDs contain a back pointer, the use of which is described below.
- Software must never schedule full-speed isochronous OUTs across an *H-Frame* boundary.
- *Case 2b*: This case can only occur for a very large isochronous IN. It is the only allowed scenario where a start-split and complete-split for the same endpoint can occur in the same micro-frame. Software must enforce this rule by scheduling the large transaction first. Large is defined to be anything larger than 579 byte maximum packet size.

A subset of the same mechanisms employed by full- and low-speed interrupt queue heads are employed in siTDs to schedule and track the portions of isochronous split transactions. The following fields are initialized by system software to instruct the host controller when to execute portions of the split transaction protocol.

- *SplitXState*. This is a single bit residing in the *Status* field of an siTD (see Figure 44-163). This bit is used to track the current state of the split transaction. The rules for managing this bit are described in Section Split Transaction Execution State Machine for Interrupt .
- *Frame S-mask*. This is a field where-in system software sets a bit corresponding to the micro-frame (within an *H-Frame*) that the host controller should execute a start-split transaction. This is always qualified by the value of the *SplitXState* bit. For example, referring to the IN example in Figure 44-162, case one, the *S-mask* would have a value of 00000001b indicating that if the siTD is traversed by the host controller, and the *SplitXState* indicates Do Start Split, and the current micro-frame as indicated by USB.FRINDEX[2:0] is 0, then execute a start-split transaction.
- *Frame C-mask.* This is a field where system software sets one or more bits corresponding to the micro-frames (within an *H-Frame*) that the host controller should execute complete-split transactions. The interpretation of this field is always qualified by the value of the *SplitXState* bit. For example, referring to the IN example

in Figure 44-162, case one, the *C-mask* would have a value of 00111100b indicating that if the siTD is traversed by the host controller, and the *SplitXState* indicates Do Complete Split, and the current micro-frame as indicated by *USB.FRINDEX*[2:0] is 2, 3, 4, or 5, then execute a complete-split transaction.

- *Back Pointer*. This field in a siTD is used to complete an IN split-transaction using the previous *H-Frame*'s siTD. This is only used when the scheduling of the complete-splits span an *H-Frame* boundary.

There exists a one-to-one relationship between a high-speed isochronous split transaction (including all start- and complete-splits) and one full-speed isochronous transaction. An siTD contains (amongst other things) buffer state and split transaction scheduling information. An siTD's buffer state always maps to one full-speed isochronous data payload. This means that for any full-speed transaction payload, a single siTD's data buffer must be used. This rule applies to both IN an OUTs. An siTD's scheduling information usually also maps to one high-speed isochronous split transaction. The exception to this rule is the *H-Frame* boundary wrap cases mentioned above.

The siTD data structure describes at most, one frame's worth of high-speed transactions and that description is strictly bounded within a frame boundary. Figure 44-163 illustrates some examples. On the top are examples of the full-speed transaction footprints for the boundary scheduling cases described above. In the middle are time-frame references for both the *B-Frames* (HS/FS/LS Bus) and the *H-Frames*. On the bottom is illustrated the relationship between the scope of an siTD description and the time references. Each *H-Frame* corresponds to a single location in the periodic frame list. The implication is that each siTD is reachable from a single periodic frame list location at a time.

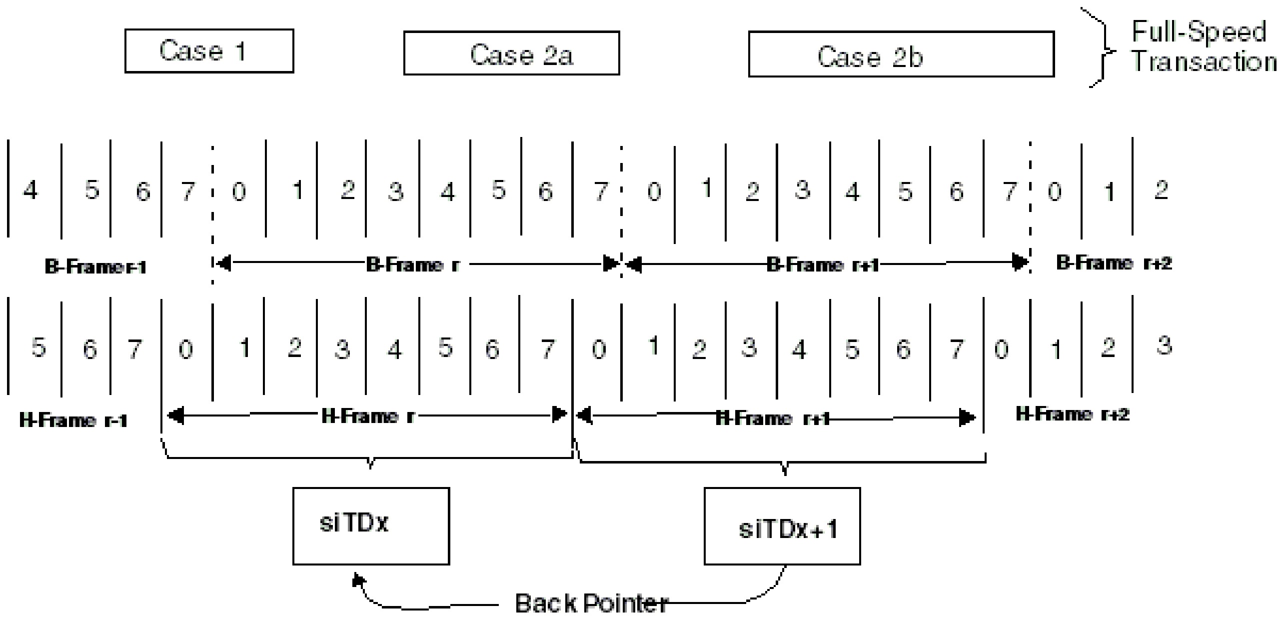


Figure 44-163. siTD Scheduling Boundary Examples

Each case is described below:

- *Case 1*: One siTD is sufficient to describe and complete the isochronous split transaction because the whole isochronous split transaction is tightly contained within a single *H-Frame*.
- *Case 2a, 2b*: Although both INs and OUTs can have these footprints, OUTs always take only one siTD to schedule. However, INs (for these boundary cases) require two siTDs to complete the scheduling of the isochronous split transaction $siTD_X$ is used to always issue the start-split and the first *N* complete-splits. The full-speed transaction (for these cases) can deliver data on the full-speed bus segment during micro-frame 7 of $H\text{-}Frame_{Y+1}$, or micro-frame 0 of $H\text{-}Frame_{Y+2}$. The complete splits are scheduled using $siTD_{X+2}$ (not shown). The complete-splits to extract this data must use the buffer pointer from $siTD_{X+1}$. The only way for the host controller to reach $siTD_{X+1}$ from $H\text{-}Frame_{Y+2}$is to use $siTD_{X+2}$'s back pointer. The host controller rules for when to use the back pointer are described is Section Periodic Isochronous - Do Complete Split .

Software must apply the following rules when calculating the schedule and linking the schedule data structures into the periodic schedule:

- Software must ensure that an isochronous split-transaction is started so that it will complete before the end of the *B-Frame*.
- Software must ensure that for a single full-speed isochronous endpoint, there is never a start-split and complete-split in *H-Frame, micro-frame 1*. This is mandated as a rule so that case 2a and case 2b can be discriminated. According to the core USB

specification, the long isochronous transaction illustrated in Case 2b, could be scheduled so that the start-split was in micro-frame 1 of *H-Frame* N and the last complete-split would need to occur in micro-frame 1 of H-Frame N+1. However, it is impossible to discriminate between cases 2a and case 2b, which has significant impact on the complexity of the host controller.

44.6.3.12.3.2 Tracking Split Transaction Progress for Isochronous Transfers

To correctly maintain the data stream, the host controller must be able to detect and report errors where device to host data is lost. Isochronous endpoints do not employ the concept of a halt on error, however the host is required to identify and report per-packet errors observed in the data stream. This includes schedule traversal problems (skipped micro-frames), timeouts and corrupted data received.

In similar kind to interrupt split-transactions, the portions of the split transaction protocol must execute in the micro-frames they are scheduled. The queue head data structure used to manage full- and low-speed interrupt has several mechanisms for tracking when portions of a transaction have occurred. Isochronous transfers use siTDs, for their transfers, and the data structures are only reachable via the schedule in the exact micro-frame in which they are required (so all the mechanism employed for tracking in queue heads is not required for siTDs). Software has the option of reusing siTD several times in the complete periodic schedule. However, it must ensure that the results of split transaction N are consumed and the siTD reinitialized (activated) before the host controller gets back to the siTD (in a future micro-frame).

Split-transaction isochronous OUTs utilize a low-level protocol to indicate which portions of the split transaction data have arrived. Control over the low-level protocol is exposed in an siTD via the fields *Transaction Position* (*TP*) and *Transaction Count* (*T-count*). If the entire data payload for the OUT split transaction is larger than 188 bytes, there will be more than one start-split transaction, each of which require proper annotation. If host hold-offs occur, then the sequence of annotations received from the host will not be complete, which is detected and handled by the transaction translator. See Section Periodic Isochronous - Do Start Split for a description on how these fields are used during a sequence of start-split transactions.

The fields *siTD.T-Count* and *siTD.TP* are used by the host controller to drive and sequence the transaction position annotations. It is the responsibility of system software to properly initialize these fields in each siTD. Once the budget for a split-transaction isochronous endpoint is established, *S-mask*, *T-Count*, and *TP* initialization values for all the siTD associated with the endpoint are constant. They remain constant until the budget for the endpoint is recalculated by software and the periodic schedule adjusted.

For IN-endpoints, the transaction translator simply annotates the response data packets with enough information to allow the host controller to identify the last data. As with split transaction Interrupt, it is the host controller's responsibility to detect when it has missed an opportunity to execute a complete-split. The following field in the siTD is used to track and detect errors in the execution of a split transaction for an IN isochronous endpoint.

- *C-prog-mask*. This is an eight-bit bit-vector where the host controller keeps track of which complete-splits have been executed. Due to the nature of the Transaction Translator periodic pipeline, the complete-splits need to be executed in-order. The host controller needs to detect when the complete-splits have not been executed in order. This can only occur due to system hold-offs where the host controller cannot get to the memory-based schedule. *C-prog-mask* is a simple bit-vector that the host controller sets a bit for each complete-split executed. The bit position is determined by the micro-frame (USB.FRINDEX[2:0]) number in which the complete-split was executed. The host controller always checks *C-prog-mask* before executing a complete-split transaction. If the previous complete-splits have not been executed, then it means one (or more) have been skipped and data has potentially been lost. System software is required to initialize this field to zero before setting an siTD's *Active* bit to a one.

If a transaction translator returns with the final data before all of the complete-splits have been executed, the state of the transfer is advanced so that the remaining complete-splits are not executed. Refer to Section Asynchronous - Do Complete Split for a description on how the state of the transfer is advanced. It is important to note that an IN siTD is retired based solely on the responses from the Transaction Translator to the complete-split transactions. This means, for example, that it is possible for a transaction translator to respond to a complete-split with an MDATA PID. The number of bytes in the MDATA's data payload could cause the siTD field *Total Bytes to Transfer* to decrement to zero. This response can occur, before all of the scheduled complete-splits have been executed. In other interface, data structures (for example, high-speed data streams through queue heads), the transition of *Total Bytes to Transfer* to zero signals the end of the transfer and results in setting of the *Active* bit to zero. However, in this case, the result has not been delivered by the Transaction Translator and the host must continue with the next complete-split transaction to extract the residual transaction state. This scenario occurs because of the pipeline rules for a Transaction Translator (see Chapter 11 of the Universal Serial Bus Revision 2.0). In summary the periodic pipeline rules require that on a micro-frame boundary, the Transaction Translator will hold the final two bytes received (if it has not seen an End Of Packet (EOP)) in the full-speed bus pipe stage and give the remaining bytes to the high-speed pipeline stage. At the micro-frame boundary, the Transaction Translator could have received the entire packet (including both CRC bytes) but not received the packet EOP. In the next micro-frame, the Transaction Translator will

respond with an MDATA and send all of the data bytes (with the two CRC bytes being held in the full-speed pipeline stage). This could cause the siTD to decrement its *Total Bytes to Transfer* field to zero, indicating it has received all expected data. The host must still execute one more (scheduled) complete-split transaction in order to extract the results of the full-speed transaction from the Transaction Translator (for example, the Transaction Translator may have detected a CRC failure, and this result must be forwarded to the host).

If the host experiences hold-offs that cause the host controller to skip one or more (but not all) scheduled split transactions for an isochronous OUT, then the protocol to the transaction translator will not be consistent and the transaction translator will detect and react to the problem. Likewise, for host hold-offs that cause the host controller to skip one or more (but not all) scheduled split transactions for an isochronous IN, the *C-prog-mask* is used by the host controller to detect errors. However, if the host experiences a hold-off that causes it to skip all of an siTD, or an siTD expires during a host hold off (for example, a hold-off occurs and the siTD is no longer reachable by the host controller in order for it to report the hold-off event), then system software must detect that the siTDs have not been processed by the host controller (that is, state not advanced) and report the appropriate error to the client driver.

44.6.3.12.3.3 Split Transaction Execution State Machine for Isochronous

In the following presentation, all references to micro-frame are in the context of a micro-frame within an *H-Frame.*

If the *Active* bit in the *Status* byte is a zero, the host controller will ignore the siTD and continue traversing the periodic schedule. Otherwise the host controller will process the siTD as specified below. A split transaction state machine is used to manage the split-transaction protocol sequence. The host controller uses the fields defined in Section Tracking Split Transaction Progress for Interrupt Transfers , plus the variable *cMicroFrameBit* defined in Section Split Transaction Execution State Machine for Interrupt to track the progress of an isochronous split transaction. Figure 44-164 illustrates the state machine for managing an siTD through an isochronous split transaction. Bold, dotted circles denote the state of the *Active* bit in the *Status* field of a siTD. The Bold, dotted arcs denote the transitions between these states. Solid circles denote the states of the split transaction state machine and the solid arcs denote the transitions between these states. Dotted arcs and boxes reference actions that take place either as a result of a transition or from being in a state.

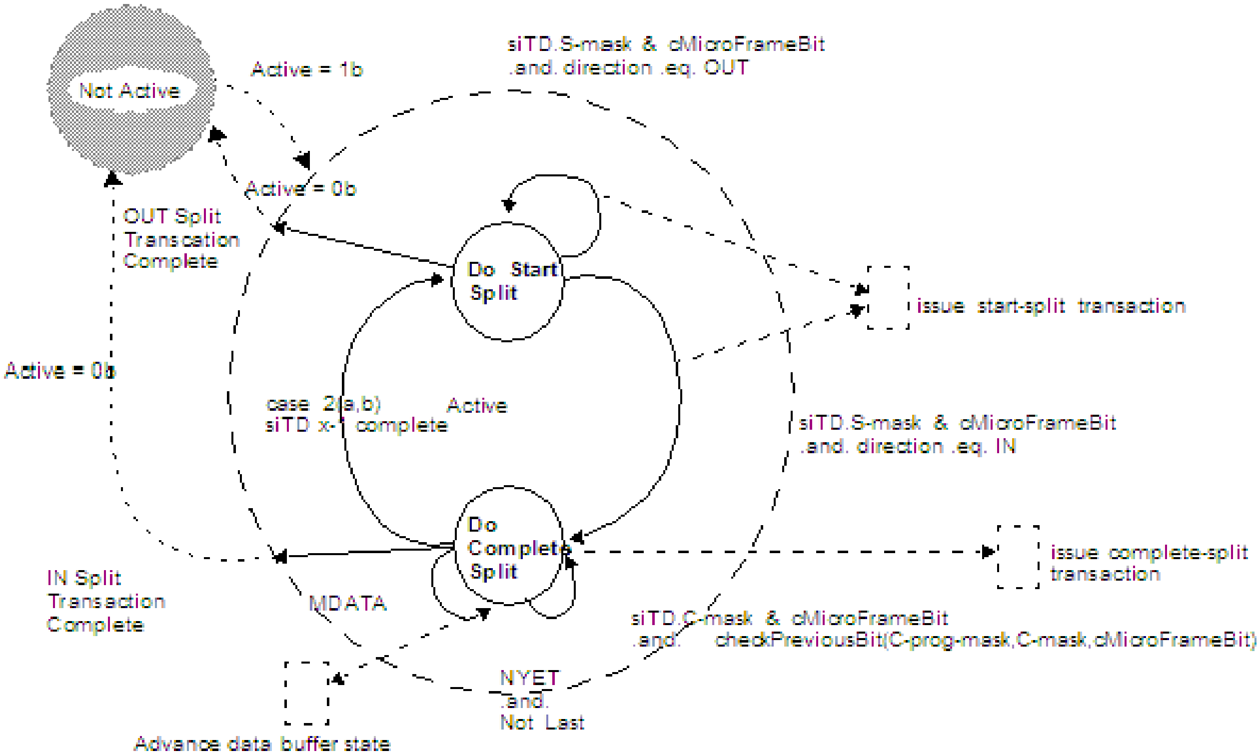


Figure 44-164. Split Transaction State Machine for Isochronous

44.6.3.12.3.4 Periodic Isochronous - Do Start Split

Isochronous split transaction OUTs use only this state. An siTD for a split-transaction isochronous IN is either initialized to this state, or the siTD transitions to this state from Do Complete Split when a case 2a (IN) or 2b scheduling boundary isochronous split-transaction completes.

Each time the host controller reaches an active siTD in this state, it checks the *siTD.S-mask* against *cMicroFrameBit*. If there is a one in the appropriate position, the siTD will execute a start-split transaction. By definition, the host controller cannot *reach* an siTD at the wrong time. If the *I/O* field indicates an IN, then the start-split transaction includes only the extended token plus the full-speed token. Software must initialize the *siTD.Total Bytes To Transfer* field to the number of bytes expected. This is usually the maximum packet size for the full-speed endpoint. The host controller exits this state when the start-split transaction is complete.

The remainder of this section is specific to an isochronous OUT endpoint (that is, the *I/O* field indicates an OUT). When the host controller executes a start-split transaction for an isochronous OUT it includes a data payload in the start-split transaction. The memory buffer address for the data payload is constructed by concatenating *siTD.Current Offset*

with the page pointer indicated by the page selector field (*siTD.P*). A zero in this field selects Page 0 and a 1 selects Page 1. During the start-split for an OUT, if the data transfer crosses a page boundary during the transaction, the host controller must detect the page cross, update the *siTD.P*-bit from a zero to a one, and begin using the *siTD.Page 1* with *siTD.Current Offset* as the memory address pointer. The field *siTD.TP* is used to annotate each start-split transaction with the indication of which part of the split-transaction data the current payload represents (ALL, BEGIN, MID, END). In all cases the host controller simply uses the value in *siTD.TP* to mark the start-split with the correct transaction position code.

T-Count is always initialized to the number of start-splits for the current frame. *TP* is always initialized to the first required transaction position identifier. The scheduling boundary case (see Figure 44-163) is used to determine the initial value of *TP*. The initial cases are summarized in Table 44-191.

Table 44-191. Initial Conditions for OUT siTD's TP and T-count Fields

Case	T-count	TP	Description
1, 2a	=1	ALL	When the OUT data payload is less than (or equal to) 188 bytes, only one start-split is required to move the data. The one start-split must be marked with an ALL.
1, 2a	!=1	BEGIN	When the OUT data payload is greater than 188 bytes more than one start-split must be used to move the data. The initial start-split must be marked with a BEGIN.

After each start-split transaction is complete, the host controller updates *T-Count* and *TP* appropriately so that the next start-split is correctly annotated.

Table 44-192 illustrates all of the *TP* and *T-count* transitions, which must be accomplished by the host controller.

.

Table 44-192. Transaction Position (TP)/Transaction Count (T-Count) Transition Table

TP	T-count next	TP next	Description
ALL	0	N/A	Transition from ALL, to done.
BEGIN	1	END	Transition from BEGIN to END. Occurs when *T-count* starts at 2.
BEGIN	!=1	MID	Transition from BEGIN to MID. Occurs when *T-count* starts at greater than 2.
MID	!=1	MID	*TP* stays at MID while *T-count* is not equal to 1 (that is, greater than 1). This case can occur for any of the scheduling boundary cases where the *T-count* starts greater than 3.
MID	1	END	Transition from MID to END. This case can occur for any of the scheduling boundary cases where the *T-count* starts greater than 2.

The start-split transactions do not receive a handshake from the transaction translator, so the host controller always advances the transfer state in the siTD after the bus transaction is complete. To advance the transfer state the following operations take place:

- The *siTD.Total Bytes To Transfer* and the *siTD.Current Offset* fields are adjusted to reflect the number of bytes transferred.
- The *siTD.P* (page selector) bit is updated appropriately.
- The *siTD.TP* and *siTD.T-count* fields are updated appropriately as defined in Table 44-192.

These fields are then written back to the memory based siTD. The *S-mask* is fixed for the life of the current budget. As mentioned above, *TP* and *T-count* are set specifically in each siTD to reflect the data to be sent from this siTD. Therefore, regardless of the value of *S-mask*, the actual number of start-split transactions depends on *T-count* (or equivalently, *Total Bytes to Transfer*). The host controller must set the *Active* bit to a zero when it detects that all of the schedule data has been sent to the bus. The preferred method is to detect when *T-Count* decrements to zero as a result of a start-split bus transaction. Equivalently, the host controller can detect when *Total Bytes to Transfer* decrements to zero. Either implementation must ensure that if the initial condition is *Total Bytes to Transfer* equal to zero and *T-count* is equal to a one, then the host controller will issue a single start-split, with a zero-length data payload. Software must ensure that *TP*, *T-count* and *Total Bytes to Transfer* are set to deliver the appropriate number of bus transactions from each siTD. An inconsistent combination will yield undefined behavior.

If the host experiences hold-offs that cause the host controller to skip start-split transactions for an OUT transfer, the state of the transfer will not progress appropriately. The transaction translator will observe protocol violations in the arrival of the start-splits for the OUT endpoint (that is, the transaction position annotation will be incorrect as received by the transaction translator).

Example scenarios are described in Section Split Transaction for Isochronous - Processing Examples .

A host controller implementation can optionally track the progress of an OUT split transaction by setting appropriate bits in the *siTD.C-prog-mask* as it executes each scheduled start-split. The *checkPreviousBit*() algorithm defined in Section Periodic Isochronous - Do Complete Split can be used prior to executing each start-split to determine whether start-splits were skipped. The host controller can use this mechanism to detect missed micro-frames. It can then set the siTD's *Active* bit to zero and stop execution of this siTD. This saves on both memory and high-speed bus bandwidth.

44.6.3.12.3.5 Periodic Isochronous - Do Complete Split

This state is only used by a split-transaction isochronous IN endpoint. This state is entered unconditionally from the Do Start State after a start-split transaction is executed for an IN endpoint. Each time the host controller visits an siTD in this state, it conducts a number of tests to determine whether it should execute a complete-split transaction. The individual tests are listed below. The sequence they are applied depends on which micro-frame the host controller is currently executing which means that the tests might not be applied until after the siTD referenced from the back pointer has been fetched.

- Test A. *cMicroFrameBit* is bit-wise anded with *siTD.C-mask* field. A non-zero result indicates that software scheduled a complete-split for this endpoint, during this micro-frame. This test is always applied to a newly fetched siTD that is in this state.
- Test B. The *siTD.C-prog-mask* bit vector is checked to determine whether the previous complete splits have been executed. An example algorithm is below (this is slightly different than the algorithm used in Section Periodic Isochronous - Do Complete Split). The sequence in which this test is applied depends on the current value of USB.FRINDEX[2:0]. If USB.FRINDEX[2:0] is 0 or 1, it is not applied until the back pointer has been used. Otherwise it is applied immediately.

```
Algorithm Boolean CheckPreviousBit(siTD.C-prog-mask, siTD.C-mask, cMicroFrameBit)
Begin
          Boolean rvalue = TRUE;
          previousBit = cMicroFrameBit rotate-right(1)
          -- Bit-wise anding previousBit with C-mask indicates whether there was an intent
          -- to send a complete split in the previous micro-frame. So, if the
          -- 'previous bit' is set in C-mask, check C-prog-mask to make sure it
          -- happened.
          if previousBit bitAND siTD.C-mask then
                    if not (previousBit bitAND siTD.C-prog-mask) then
                               rvalue = FALSE
                    End if
          End if
          Return rvalue
End Algorithm
```

If Test A is true and USB.FRINDEX[2:0] is zero or one, then this is a case 2a or 2b scheduling boundary (see Figure 44-162). See Section Periodic Isochronous - Do Complete Split for details in handling this condition.

If Test A and Test B evaluate to true, then the host controller will execute a complete-split transaction using the transfer state of the current siTD. When the host controller commits to executing the complete-split transaction, it updates *QH.C-prog-mask* by bit-ORing with *cMicroFrameBit*. The transfer state is advanced based on the completion status of the complete-split transaction. To advance the transfer state of an IN siTD, the host controller must:

- Decrement the number of bytes received from *siTD.Total Bytes To Transfer,*
- Adjust *siTD.Current Offset* by the number of bytes received,

- Adjust *siTD.P* (page selector) field if the transfer caused the host controller to use the next page pointer, and
- Set any appropriate bits in the *siTD.Status* field, depending on the results of the transaction.

Note that if the host controller encounters a condition where *siTD.Total Bytes To Transfer* is zero, and it receives more data, the host controller must not write the additional data to memory. The *siTD.Status.Active* bit must be set to zero and the *siTD.Status.Babble Detected* bit must be set to a one. The fields *siTD.Total Bytes To Transfer*, *siTD.Current Offset*, and *siTD.P* (page selector) are not required to be updated as a result of this transaction attempt.

The host controller must accept (assuming good data packet CRC and sufficient room in the buffer as indicated by the value of *siTD.Total Bytes To Transfer*) MDATA and DATA0/1 data payloads up to and including 192 bytes. A host controller implementation may optionally set *siTD.Status Active* to a zero and *siTD.Status.Babble Detected* to a one when it receives and MDATA or DATA0/1 with a data payload of more than 192 bytes. The following responses have the noted effects:

- ERR. The full-speed transaction completed with a time-out or bad CRC and this is a reflection of that error to the host. The host controller sets the *ERR* bit in the *siTD.Status* field and sets the *Active* bit to a zero.
- Transaction Error (XactErr). The complete-split transaction encounters a Timeout, CRC16 failure, etc. The *siTD.Status* field *XactErr* field is set to a one and the complete-split transaction must be retried immediately. The host controller must use an internal error counter to count the number of retries as a counter field is not provided in the siTD data structure. The host controller will not retry more than two times. If the host controller exhausts the retries or the end of the micro-frame occurs, the *Active* bit is set to zero.
- DATAx (0 or 1). This response signals that the final data for the split transaction has arrived. The transfer state of the siTD is advanced and the *Active* bit is set to a zero. If the *Bytes To Transfer* field has not decremented to zero (including the reception of the data payload in the DATAx response), then less data than was expected, or allowed for was actually received. This *short packet* event does not set the USBINT status bit in the USBSTS register to a one. The host controller will not detect this condition.
- NYET (and Last). On each NYET response, the host controller also checks to determine whether this is the last complete-split for this split transaction. Last was defined in Section Periodic Isochronous - Do Complete Split . If it is the last complete-split (with a NYET response), then the transfer state of the siTD is not advanced (never received any data) and the *Active* bit is set to a zero. No bits are set in the *Status* field because this is essentially a skipped transaction. The transaction translator must have responded to all the scheduled clompete-splits with NYETs,

meaning that the start-split issued by the host controller was not received. This result should be interpreted by system software as if the transaction was completely skipped. The test for whether this is the last complete split can be performed by XORing C-mask with C-prog-mask. A zero result indicates that all complete-splits have been executed.

- MDATA (and Last). See above description for testing for Last. This can only occur when there is an error condition. Either there has been a babble condition on the full-speed link, which delayed the completion of the full-speed transaction, or software set up the *S-mask* and/or *C-mask*s incorrectly. The host controller must set *XactErr* bit to a one and the *Active* bit is set to a zero.
- NYET (and not Last). See above description for testing for Last. The complete-split transaction received a NYET response from the transaction translator. Do not update any transfer state (except for *C-prog-mask*) and stay in this state.
- MDATA (and not Last). The transaction translator responds with an MDATA when it has partial data for the split transaction. For example, the full-speed transaction data payload spans from micro-frame X to X+1 and during micro-frame X, the transaction translator will respond with an MDATA and the data accumulated up to the end of micro-frame X. The host controller advances the transfer state to reflect the number of bytes received.

If Test A succeeds, but Test B fails, it means that one or more of the complete-splits have been skipped. The host controller sets the *Missed Micro-Frame* status bit and sets the *Active* bit to a zero.

44.6.3.12.3.6 Complete-Split for Scheduling Boundary Cases 2a, 2b

Boundary cases 2a and 2b (INs only) (see Figure 44-162) require that the host controller use the transaction state context of the previous siTD to finish the split transaction. Table 44-193 enumerates the transaction state fields.

Table 44-193. Summary siTD Split Transaction State

Buffer State	Status	Execution Progress
Total Bytes To Transfer P (page select) Current Offset TP (transaction position) T-count (transaction count)	All bits in the status field	C-prog-mask

NOTE

TP and *T-count* are used only for Host to Device (OUT) endpoints.

If software has budgeted the schedule of this data stream with a frame wrap case, then it must initialize the *siTD.Back Pointer* field to reference a valid siTD and will have the *siTD.Back Pointer.T-bit* in the *siTD.Back Pointer*

field set to a zero. Otherwise, software must set the *siTD.Back Pointer.T-bit* in the *siTD.Back Pointer* field to a one. The host controller's rules for interpreting when to use the *siTD.Back Pointer* field are listed below. These rules apply only when the siTD's *Active* bit is a one and the *SplitXState* is Do Complete Split.

- When cMicroFrameBit is a 1h and the siTDX.Back Pointer.T-bit is a zero, or
- If cMicroFrameBit is a 2h and siTDX.S-mask[0] is a zero

When either of these conditions apply, then the host controller must use the transaction state from $siTD_{X-1}$.

In order to access $siTD_{X-1}$, the host controller reads on-chip the siTD referenced from $siTD_{X}$*.Back Pointer*.

The host controller must save the entire state from $siTD_{X}$ while processing $siTD_{X-1}$. This is to accommodate for case 2b processing. The host controller must not recursively walk the list of *siTD.Back Pointers*.

If $siTD_{X-1}$ is active (*Active* bit is a one and *SplitXStat* is Do Complete Split), then both Test A and Test B are applied as described above. If these criteria to execute a complete-split are met, the host controller executes the complete split and evaluates the results as described above. The transaction state (see Table 44-193) of $siTD_{X-1}$ is appropriately advanced based on the results and written back to memory. If the resultant state of $siTD_{X-1}$'s *Active* bit is a one, then the host controller returns to the context of $siTD_{X}$, and follows its next pointer to the next schedule item. No updates to $siTD_{X}$ are necessary.

If $siTD_{X-1}$ is active (*Active* bit is a one and *SplitXStat* is Do Start Split), then the host controller must set *Active* bit to a zero and *Missed Micro-Frame* status bit to a one and the resultant status written back to memory.

If $siTD_{X-1}$*'sActive* bit is a zero, (because it was zero when the host controller first visited $siTD_{X-1}$ via $siTD_{X}$*'s* back pointer, it transitioned to zero as a result of a detected error, or the results of $siTD_{X-1}$*'s* complete-split transaction transitioned it to zero), then the host controller returns to the context of $siTD_{X}$ and transitions its *SplitXState* to Do Start Split. The host controller then determines whether the case 2b start split boundary condition exists (that is, if *cMicroframeBit* is a 1b and $siTD_{X}$*.S-mask*[0] is a 1b). If this criterion is met the host controller immediately executes a start-split transaction and appropriately advances the transaction state of $siTD_{X}$, then follows $siTD_{X}$*.Next Pointer* to the next schedule item. If the criterion is not met, the host controller simply follows $siTD_{X}$*.Next Pointer* to the next schedule item. Note that in the case of a 2b boundary case, the split-transaction of $siTD_{X-1}$ will have its *Active* bit set to zero when the host controller returns

to the context of *siTD*$_X$. Also, note that software should not initialize an siTD with *C-mask* bits 0 and 1 set to a one and an *S-mask* with bit zero set to a one. This scheduling combination is not supported and the behavior of the host controller is undefined.

44.6.3.12.3.7 Split Transaction for Isochronous - Processing Examples

There is an important difference between how the hardware/software manages the isochronous split transaction state machine and how it manages the asynchronous and interrupt split transaction state machines. The asynchronous and interrupt split transaction state machines are encapsulated within a single queue head. The progress of the data stream depends on the progress of each split transaction. In some respects, the split-transaction state machine is sequenced via the Execute Transaction queue head traversal state machine (see Figure 44-155).

Isochronous is a pure time-oriented transaction/data stream. The interface data structures are optimized to efficiently describe transactions that need to occur at specific times. The isochronous split-transaction state machine must be managed across these time-oriented data structures. This means that system software must correctly describe the scheduling of split-transactions across more than one data structure.

Then the host controller must make the appropriate state transitions at the appropriate times, in the correct data structures.

For example, Table 44-194 illustrates a couple of frames worth of scheduling required to schedule a case 2a full-speed isochronous data stream.

Table 44-194. Example Case 2a - Software Scheduling siTDs for an IN Endpoint

siTDX		Micro-Frames								Initial
#	Masks	0	1	2	3	4	5	6	7	SplitXState
X	S-Mask	-	-	-	-	1	-	-	-	Do Start Split
	C-Mask	1	1	-	-	-	-	1	1	
X+1	S-Mask	-	-	-	-	1	-	-	-	Do Complete Split
	C-Mask	1	1					1	1	
X+2	S-Mask	-	-	-	-	1	-	-	-	Do Complete Split
	C-Mask	1	1					1	1	
X+3	S-Mask	Repeats previous pattern								Do Complete Split
	C-Mask									

This example shows the first three siTDs for the transaction stream. Because this is the case-2a frame-wrap case, *S-masks* of all siTDs for this endpoint have a value of 10h (a one bit in micro-frame 4) and *C-mask* value of C3h (one-bits in micro-frames 0,1, 6 and 7). Additionally, sofware ensures that the *Back Pointer* field of each siTD references the appropriate siTD data structure (and the *Back PointerT-bits* are set to zero).

The initial *SplitXState* of the first siTD is Do Start Split. The host controller will visit the first siTD eight times during frame X. The C-mask bits in micro-frames 0 and 1 are ignored because the state is Do Start Split. During micro-frame 4, the host controller determines that it can run a start-split (and does) and changes *SplitXState* to Do Complete Split. During micro-frames 6 and 7, the host controller executes complete-splits. Notice the siTD for frame X+1 has its *SplitXState* initialized to Do Complete Split. As the host controller continues to traverse the schedule during *H-Frame* X+1, it will visit the second siTD eight times. During micro-frames 0 and 1 it will detect that it must execute complete-splits.

During *H-Frame* X+1, micro-frame 0, the host controller detects that $siTD_{X+1}$'s *Back Pointer.T-bit* is a zero, saves the state of $siTD_{X+1}$ and fetches $siTD_X$. It executes the complete split transaction using the transaction state of $siTD_X$. If the $siTD_X$ split transaction is complete, siTD's *Active* bit is set to zero and results written back to $siTD_X$. The host controller retains the fact that $siTD_X$ is retired and transitions the *SplitXState* in the $siTD_{X+1}$ to Do Start Split. At this point, the host controller is prepared to execute the start-split for $siTD_{X+1}$ when it reaches micro-frame 4. If the split-transaction completes early (transaction-complete is defined in Section Periodic Isochronous - Do Complete Split), that is, before all the scheduled complete-splits have been executed, the host controller will transition *$siTD_X$.SplitXState* to Do Start Split early and naturally skip the remaining scheduled complete-split transactions. For this example, $siTD_{X+1}$ does not receive a DATA0 response until *H-Frame* X+2, micro-frame 1.

During *H-Frame X+2,* micro-frame 0, the host controller detects that $siTD_{X+2}$'s *Back Pointer.T-bit* is a zero, saves the state of $siTD_{X+2}$ and fetches $siTD_{X+1}$. As described above, it executes another split transaction, receives an MDATA response, updates the transfer state, but does not modify the *Active* bit. The host controller returns to the context of $siTD_{X+2}$, and traverses its next pointer without any state change updates to $siTD_{X+2}$. S

During *H-Frame X+2*, micro-frame 1, the host controller detects $siTD_{X+2}$'s *S-mask[0]* is a zero, saves the state of $siTD_{X+2}$ and fetches $siTD_{X+1}$. It executes another complete-split transaction, receives a DATA0 response, updates the transfer state and sets the *Active* bit to a zero. It returns to the state of $siTD_{X+2}$ and changes its *SplitXState* to Do Start Split. At this point, the host controller is prepared to execute start-splits for $siTD_{X+2}$ when it

reaches micro-frame 4. <TBD describe how software detects that there was missing micro-frames (don't think we care about missing out micro-frames. There is enough residual state to identify than not all transactions were executed.).

44.6.3.13 Host Controller Pause

When the host controller's *HCHalted* bit in the USBSTS register is a zero, the host controller is sending SOF (Start OF Frame) packets down all enabled ports. When the schedules are enabled, the EHCI host controller will access the schedules in main memory each micro-frame. This constant pinging of main memory is known to create ARM platform power management problems for mobile systems. Specifically, mobile systems aggressively manage the state of the ARM platform, based on recent history usage. In the more aggressive power saving modes, the ARM platform can disable its caches. Current PC architectures assume that bus-master accesses to main memory must be cache-coherent. So, when bus masters are busy touching memory, the ARM platform power management software can detect this activity over time and inhibit the transition of the ARM platform into its lowest power savings mode. USB controllers are bus-masters and the frequency at which they access their memory-based schedules keeps the ARM platform power management software from placing the ARM platform into its lowest power savings state.

USB Host controllers don't access main memory when they are suspended. However, there are a variety of reasons why placing the USB controllers into suspend won't work, but they are beyond the scope of this document. The base requirement is that the USB controller needs to be kept out of main memory, while at the same time, the USB bus is kept from going into suspend.

EHCI controllers provide a large-grained mechanism that can be manipulated by system software to change the memory access pattern of the host controller. System software can manipulate the schedule enable bits in the USBCMD register to turn on/off the scheduling traversal. A software heuristic can be applied to implement an on/off duty cycle that allows the USB to make reasonable progress and allow the ARM platform power management to get the ARM platform into its lowest power state. This method is not intended to be applied at all times to throttle USB, but should only be applied in very specific configurations and usage loads. For example, when only a keyboard or mouse is attached to the USB, the heuristic could detect times when the USB is attempting to move data only very infrequently and can adjust the duty cycle to allow the ARM platform to reach its low power state for longer periods of time. Similarly, it could detect increases in the USB load and adjust the duty cycle appropriately, even to the point where the schedules are never disabled. The assumption here is that the USB is moving data and the ARM platform will be required to process the data streams.

It is suggested that in order to provide a complete solution for the system, the companion host controllers should also provide a similar method to allow system software to inhibit the companion host controller from accessing its shared memory based data structures (schedule lists or otherwise).

44.6.3.14 Port Test Modes -Host Operational Model

EHCI host controllers must implement the port test modes Test J_State, Test K_State, Test_Packet, Test Force_Enable, and Test SE0_NAK as described in the USB Specification Revision 2.0. The system is only allowed to test ports that are owned by the EHCI controller (for example, *CF-bit* is a one and *PortOwner* bit is a zero). System software is allowed to have at most one port in test mode at a time. Placing more than one port in test mode will yield undefined results. The required, per port test sequence is (assuming the *CF-bit* in the USB.CONFIGFLAG register is a one):

- Disable the periodic and asynchronous schedules by setting the *Asynchronous Schedule Enable* and *Periodic Schedule Enable* bits in the USBCMD register to a zero.
- Place all enabled root ports into the suspended state by setting the *Suspend* bit in each appropriate USB.PORTSC register to a one.
- Set the *Run/Stop* bit in the USBCMD register to a zero and wait for the *HCHalted* bit in the USBSTS register, to transition to a one. Note that an EHCI host controller implementation may optionally allow port testing with the *Run/Stop* bit set to a one. However, all host controllers must support port testing with *Run/Stop* set to a zero and *HCHalted* set to a one.
- Set the *Port Test Control* field in the port under test PORTSC register to the value corresponding to the desired test mode. If the selected test is Test_Force_Enable, then the *Run/Stop* bit in the USBCMD register must then be transitioned back to one, in order to enable transmission of SOFs out of the port under test.
- When the test is complete, system software must ensure the host controller is halted (*HCHalted* bit is a one) then it terminates and exits test mode by setting *HCReset* to a one.

44.6.3.15 Interrupts-Host Operational Model

The EHCI Host Controller hardware provides interrupt capability based on a number of sources. There are several general groups of interrupt sources:

- Interrupts as a result of executing transactions from the schedule (success and error conditions),

- Host controller events (Port change events, etc.), and
- Host Controller error events

All transaction-based sources are maskable through the Host Controller's Interrupt Enable register (USBINTR, see Section Interrupt Enable Register (USB_USBINTR)). Additionally, individual transfer descriptors can be marked to generate an interrupt on completion. This section describes each interrupt source and the processing that occurs in response to the interrupt.

During normal operation, interrupts may be immediate or deferred until the next interrupt threshold occurs. The interrupt threshold is a tunable parameter via the *Interrupt Threshold Control* field in the USBCMD register. The value of this register controls when the host controller will generate an interrupt on behalf of normal transaction execution. When a transaction completes during an interrupt interval period, the interrupt signaling the completion of the transfer will not occur until the interrupt threshold occurs. For example, the default value is eight micro-frames. This means that the host controller will not generate interrupts any more frequently than once every eight micro-frames.

Section Host System Error details effects of a host system error.

If an interrupt has been scheduled to be generated for the current interrupt threshold interval, the interrupt is not signaled until after the status for the last complete transaction in the interval has been written back to host memory. This may sometimes result in the interrupt not being signaled until the next interrupt threshold.

Initial interrupt processing is the same, regardless of the reason for the interrupt. When an interrupt is signaled by the hardware, ARM platform control is transferred to host controller's USB interrupt handler. The precise mechanism to accomplish the transfer is OS specific. For this discussion it is just assumed that control is received. When the interrupt handler receives control, its first action is to reads the USBSTS (USB Status Register). It then acknowledges the interrupt by clearing all of the interrupt status bits by writing ones to these bit positions. The handler then determines whether the interrupt is due to schedule processing or some other event. After acknowledging the interrupt, the handler (via an OS-specific mechanism), schedules a deferred procedure call (DPC) which will execute later. The DPC routine processes the results of the schedule execution. The precise mechanisms used are beyond the scope of this document.

Note: the host controller is not required to de-assert a currently active interrupt condition when software sets the interrupt enables (in the USBINR register, see Section Interrupt Enable Register (USB_USBINTR)) to a zero. The only reliable method software should use for acknowledging an interrupt is by transitioning the appropriate status bits in the USBSTS register (Section USB Status Register (USB_USBSTS)) from a one to a zero.

44.6.3.15.1 Transfer/Transaction Based Interrupts

These interrupt sources are associated with transfer and transaction progress. They are all dependent on the next interrupt threshold.

44.6.3.15.1.1 Transaction Error

A transaction error is any error that caused the host controller to think that the transfer did not complete successfully. Table 44-195 lists the events/responses that the host can observe as a result of a transaction. The effects of the error counter and interrupt status are summarized in the following paragraphs. Most of these errors set the *XactErr* status bit in the appropriate interface data structure.

There is a small set of protocol errors that relate only when executing a queue head and fit under the umbrella of a WRONG PID error that are significant to explicitly identify. When these errors occur, the *XactErr* status bit in the queue head is set and the *CErr* field is decremented. When the *PIDCode* indicates a SETUP, the following responses are protocol errors and result in *XactErr* bit being set to a one and the *CErr* field being decremented.

- *EPS* field indicates a high-speed device and it returns a Nak handshake to a SETUP.
- *EPS* field indicates a high-speed device and it returns a Nyet handshake to a SETUP.
- *EPS* field indicates a low- or full-speed device and the complete-split receives a Nak handshake.

Table 44-195. Summary of Transaction Errors

Event / Result	Queue Head/qTD/iTD/siTD Side-effects		USB Status Register (USBSTS)
	Cerr	Status Field	USBERRINT
CRC	-1	XactErr set to a one.	1[1]
Timeout	-1	XactErr set to a one.	1[1]
Bad PID[2]	-1	XactErr set to a one.	1[1]
Babble	N/A	Section Serial Bus Babble	1
Buffer Error	N/A	Section Data Buffer Error	

1. If occurs in a queue head, then *USBERRINT* is asserted only when *CErr* counts down from a one to a zero. In addition the queue is halted, see Halting a Queue Head .
2. The host controller received a response from the device, but it could not recognize the PID as a valid PID.

44.6.3.15.1.2 Serial Bus Babble

When a device transmits more data on the USB than the host controller is expecting for this transaction, it is defined to be babbling. In general, this is called a *Packet Babble*. When a device sends more data than the *Maximum Length* number of bytes, the host controller sets the *Babble Detected* bit to a one and halts the endpoint if it is using a

queue head (see Halting a Queue Head). *Maximum Length* is defined as the minimum of *Total Bytes to Transfer* and *Maximum Packet Size*. The *CErr* field is not decremented for a packet babble condition (only applies to queue heads). A babble condition also exists if IN transaction is in progress at High-speed EOF2 point. This is called a frame babble. A frame babble condition is recorded into the appropriate schedule data structure. In addition, the host controller must disable the port to which the frame babble is detected.

The *USBERRINT* bit in the USB.USBSTS register is set to a one and if the *USB Error Interrupt Enable* bit in the USB.USBINTR register is a one, then a hardware interrupt is signaled to the system at the next interrupt threshold. The host controller must never start an OUT transaction that will babble across a micro-frame EOF.

NOTE

When a host controller detects a data PID mismatch, it must either: disable the packet babble checking for the duration of the bus transaction or do packet babble checking based solely on *Maximum Packet Size*. The USB core specification defines the requirements on a data receiver when it receives a data PID mismatch (for example, expects a DATA0 and gets a DATA1 or visa-versa). In summary, it must ignore the received data and respond with an ACK handshake, in order to advance the transmitter's data sequence.

The EHCI interface allows System software to provide buffers for a Control, Bulk or Interrupt IN endpoint that are not an even multiple of the maximum packet size specified by the device. Whenever a device misses an ACK for an IN endpoint, the host and device are out of synchronization with respect to the progress of the data transfer. The host controller may have advanced the transfer to a buffer that is less than maximum packet size. The device will re-send its maximum packet size data packet, with the original data PID, in response to the next IN token. In order to properly manage the bus protocol, the host controller must disable the packet babble check when it observes the data PID mismatch.

44.6.3.15.1.3 Data Buffer Error

This event indicates that an overrun of incoming data or a underrun of outgoing data has occurred for this transaction. This would generally be caused by the host controller not being able to access required data buffers in memory within necessary latency requirements. These conditions are not considered transaction errors, and do not effect

the error count in the queue head. When these errors do occur, the host controller records the fact the error occurred by setting the *Data Buffer Error* bit in the queue head, iTD or siTD.

If the data buffer error occurs on a non-isochronous IN, the host controller will not issue a handshake to the endpoint. This will force the endpoint to resend the same data (and data toggle) in response to the next IN to the endpoint.

If the data buffer error occurs on an OUT, the host controller must corrupt the end of the packet so that it cannot be interpreted by the device as a good data packet. Simply truncating the packet is not considered acceptable. An acceptable implementation option is to 1's complement the CRC bytes and send them. There are other options suggested in the Transaction Translator section of the USB Specification Revision 2.0.

44.6.3.15.1.4 USB Interrupt (Interrupt on Completion (IOC))

Transfer Descriptors (iTDs, siTDs, and queue heads (qTDs)) contain a bit that can be set to cause an interrupt on their completion. The completion of the transfer associated with that schedule item causes the USB Interrupt (USBINT) bit in the USB.USBSTS register to be set to a one. In addition, if a short packet is encountered on an IN transaction associated with a queue head, then this event also causes USBINT to be set to a one. If the USB Interrupt Enable bit in the USB.USBINTR register is set to a one, a hardware interrupt is signaled to the system at the next interrupt threshold. If the completion is because of errors, the *USBERRINT* bit in the USB.USBSTS register is also set to a one.

44.6.3.15.1.5 Short Packet

Reception of a data packet that is less than the endpoint's Max Packet size during Control, Bulk or Interrupt transfers signals the completion of the transfer. Whenever a short packet completion occurs during a queue head execution, the *USBINT* bit in the USB.USBSTS register is set to a one. If the *USB Interrupt Enable* bit is set in the USB.USBINTR register, a hardware interrupt is signaled to the system at the next interrupt threshold.

44.6.3.15.2 Host Controller Event Interrupts

These interrupt sources are independent of the interrupt threshold (with the one exception being the Interrupt on Async Advance, see Section Interrupt on Async Advance).

44.6.3.15.2.1 Port Change Events

Port registers contain status and status change bits. When the status change bits are set to a one, the host controller sets the *Port Change Detect* bit in the USBSTS register to a one. If the *Port Change Interrupt Enable* bit in the USB.USBINTR register is a one, then the host controller will issue a hardware interrupt. The port status change bits include:

- Connect Status Change
- Port Enable/Disable Change
- Over-current Change
- Force Port Resume

44.6.3.15.2.2 Frame List Rollover

This event indicates that the host controller has wrapped the frame list. The current programmed size of the frame list effects how often this interrupt occurs. If the frame list size is 1024, then the interrupt will occur every 1024 milliseconds, if it is 512, then it will occur every 512 milliseconds, etc. When a frame list rollover is detected, the host controller sets the *Frame List Rollover* bit in the USB.USBSTS register to a one. If the *Frame List Rollover Enable* bit in the USB.USBINTR register is set to a one, the host controller issues a hardware interrupt. This interrupt is not delayed to the next interrupt threshold.

44.6.3.15.2.3 Interrupt on Async Advance

This event is used for deterministic removal of queue heads from the asynchronous schedule. Whenever the host controller advances the on-chip context of the asynchronous schedule, it evaluates the value of the *Interrupt on Async Advance Doorbell* bit in the USB.USBCMD register. If it is a one, it sets the *Interrupt on Async Advance* bit in the USB.USBSTS register to a one. If the *Interrupton Async Advance Enable* bit in the USB.USBINTR register is a one, the host controller issues a hardware interrupt at the next interrupt threshold. A detailed explanation of this feature is described in Section Removing Queue Heads from Asynchronous Schedule .

44.6.3.15.2.4 Host System Error

The host controller is a bus master and any interaction between the host controller and the system may experience errors. The type of host error may be catastrophic to the host controller (such as a Master Abort) making it impossible for the host controller to continue in a coherent fashion. In the presence of non-catastrophic host errors, such as parity errors, the host controller could potentially continue operation. The recommended behavior for these types of errors is to escalate it to a catastrophic error and halt the host controller. Host-based error must result in the following actions:

- The *Run/Stop* bit in the USB.USBCMD register is set to a zero.
- The following bits in the USB.USBSTS register are set:
  - *Host System Error* bit is to a one.
  - *HCHalted* bit is set to a one.
- If the *Host System Error Enable* bit in the USB.USBINTR register is a one, then the host controller will issue a hardware interrupt. This interrupt is not delayed to the next interrupt threshold. Table 44-196 summarizes the required actions taken on the various host errors.

Table 44-196. Summary Behavior of EHCI Host Controller on Host System Errors

Cycle Type	Master Abort	Target Abort	Data Phase Parity
Frame list pointer fetch (read)	Fatal	Fatal	Fatal [o]
siTD fetch (read)	Fatal	Fatal	Fatal [o]
siTD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
iTD fetch (read)	Fatal	Fatal	Fatal [o]
iTD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
qTD fetch (read)	Fatal	Fatal	Fatal [o]
qHD status write-back (write)	Fatal [o]	Fatal [o]	Fatal [o]
Data write	Fatal [o]	Fatal [o]	Fatal [o]
Data read	Fatal	Fatal	Fatal [o]

Potentially, a host controller implementation could continue operation without a halt. However, the recommended behavior is to halt the host controller.

NOTE

After a *Host System Error*, Software must reset the host controller through *HCReset* in the USB.USBCMD register before re-initializing and restarting the host controller.

44.6.4 EHCI Deviation

For the purposes a dual-role Host/Device controller with support for On-The-Go applications, it is necessary to deviate from the EHCI specificationEnhanced Host Controller Interface Specification for Universal Serial Bus, Revision 0.95, November 2000, Intel Corporation. http://www.intel.com. Device operation & On-The-Go operation is not specified in the EHCI and thus the implementation supported in this core is proprietary. The host mode operation of the core is near EHCI compatible with few minor differences documented in this section.

The particulars of the deviations occur in the areas summarized here:

- Embedded Transaction Translator - Allows direct attachment of FS and LS devices in host mode without the need for a companion controller.
- Device operation - In host mode the device operational registers are generally disabled and thus device mode is mostly transparent when in host mode. However, there are a couple exceptions documented in the following sections.
- Embedded design interface - This core does not a PCI Interface and therefore the PCI configuration registers described in the EHCI specification are not applicable.
- On-The-Go Operation - This design includes an On-The-Go controller for Port #1.

44.6.4.1 Embedded Transaction Translator Function

The USB-HS OTG High-Speed USB On-The-Go OTG controller supports directly connected full and low speed devices without requiring a companion controller by including the capabilities of a USB 2.0 high speed hub transaction trasnslator. Although there is no separate Tranaction Translator block in the system, the transaction translator function normally associated with a high speed hub has been implemented within the DMA and Protocol engine blocks. The embedded transaction translator fuction is an extension to EHCI interface, but makes use of the standard data structures and operational models that exist in the EHCI specification to support full and low speed devices.

44.6.4.1.1 Capability Registers

The following additions have been added to the capability registers to support the embedded Transaction Translator Function:

- N_TT added to USB.HCSPARAMS - Host Control Structural Parameters
- N_PTT added to USB.HCSPARAMS - Host Control Structural Parameters

44.6.4.1.2 Operational Registers

The following additions have been added to the operational registers to support the embedded TT:

- Addition of two-bit Port Speed (PSPD) to the Port Status & Control (USB_PORTSC1) register.

44.6.4.1.3 Discovery-EHCI Deviation

In a standard EHCI controller design, the EHCI host controller driver detects a Full speed (FS) or Low speed (LS) device by noting if the port enable bit is set after the port reset operation. The port enable will only be set in a standard EHCI controller implementation after the port reset operation and when the host and device negotiate a High-Speed connection (that is, Chirp completes successfully).

Because this controller has an embedded Transaction Translator, the port enable will always be set after the port reset operation regardless of the result of the host device chirp result and the resulting port speed will be indicated by the PSPD field in USB.PORTSCx.

Therefore, the standard EHCI host controller driver requires an alteration to handle directly connected Full and Low speed devices or hubs.

The change is a fundamental one in that is summarized in Table 44-197.

Table 44-197. Summary of EHCI

Standard EHCI	EHCI with embedded Transaction Translator
After port enable bit is set following a connection and reset sequence, the device/hub is assumed to be HS.	After port enable bit is set following a connection and reset sequence, the device/hub speed is noted from USB.PORTSCx.
FS and LS devices are assumed to be downstream from a HS hub thus, all port-level control is performed through the Hub Class to the nearest Hub.	FS and LS device can be either downstream from a HS hub or directly attached. When the FS/LS device is downstream from a HS hub, then port-level control is done using the Hub Class through the nearest Hub. When a FS/LS device is directly attached, then port-level control is accomplished using USB.PORTSCx.
FS and LS devices are assumed to be downstream from a HS hub with HubAddr=X. [where HubAddr > 0 and HubAddr is the address of the Hub where the bus transitions from HS to FS/LS (ie. Split target hub)]	FS and LS device can be either downstream from a HS hub with HubAddr = X [HubAddr > 0] or directly attached [where HubAddr = 0 and HubAddr is the address of the Root Hub where the bus transitions from HS to FS/LS (ie. Split target hub is the root hub)]

44.6.4.1.4 Data Structures

The same data structures used for FS/LS transactions though a HS hub are also used for transactions through the Root Hub with an embedded Transaction Translator. Here it is demonstrated how the Hub Address and Endpoint Speed fields should be set for directly attached FS/LS devices and hubs:

1. QH (for direct attach FS/LS) - Async. (Bulk/Control Endpoints) Periodic (Interrupt)
    - Hub Address = 0
    - Transactions to direct attached device/hub.
        - QH.EPS = Port Speed
    - Transactions to a device downstream from direct attached FS hub.
        - QH.EPS = Downstream Device Speed

NOTE

When QH.EPS = 01 (LS) and PORTSCx.PSPD = 00 (FS), a LS-pre-pid will be sent before the transmitting LS traffic.

Maximum Packet Size must be less than or equal 64 or undefined behaviour may result.

2. siTD (for direct attach FS) - Periodic (ISO Endpoint)
   - All FS ISO transactions:
     - Hub Address = 0
     - siTD.EPS = 00 (full speed)
       - Maximum Packet Size must less than or equal to 1023 or undefined behaviour may result.

44.6.4.1.5 Operational Model

The operational models are well defined for the behavior of the Transaction Translator (see USB 2.0 specificationUniversal Serial Bus Specification, Revision 2.0, April 2000, Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips. http://www.usb.org) and for the EHCI controller moving packets between system memory and a USB-HS hub. Because the embedded Transaction Translator exists within the host controller there is no physical bus between EHCI host controller driver and the USB FS/LS bus. These sections will briefly discuss the operational model for how the EHCI and Transaction Translator operational models are combined without the physical bus between. The following sections assume the reader is familiar with both the EHCI and USB 2.0 Transaction Translator operational models.

44.6.4.1.5.1 Micro- frame Pipeline

The EHCI operational model uses the concept of H-frames and B-frames to describe the pipeline between the Host (H) and the Bus (B). The embedded Transaction Translator shall use the same pipeline algorithms specified in the USB 2.0 specification for a Hub-based Transaction Translator.

All periodic transfers always begin at B-frame 0 (after SOF) and continue until the stored periodic transfers are complete. As an example of the micro-frame pipeline implemented in the embedded Transaction Translator, all periodic transfers that are tagged in EHCI to execute in H-frame 0 will be ready to execute on the bus in B-frame 0.

It is important to note that when programming the S-mask and C-masks in the EHCI data structures to schedule periodic transfers for the embedded Transaction Translator, the EHCI host controller driver must follow the same rules specified in EHCI for programming the S-mask and C-mask for downstream Hub-based Transaction Translators.

Once periodic transfers are exhausted, any stored asynchronous transfer will be moved. Asynchronous transfers are opportunistic in that they shall execute whenever possible and their operation is not tied to H-frame and B-frame boundaries with the exception that an asynchronous transfer can not babble through the SOF (start of B-frame 0.)

44.6.4.1.5.2 Split State Machines

The start and complete split operational model differs from EHCI slightly because there is no bus medium between the EHCI controller and the embedded Transaction Translator. Where a start or complete-split operation would occur by requesting the split to the HS hub, the start/complete split operation is simple an internal operation to the embedded Transaction Translator. Table 44-198 summarizes the conditions where handshakes are emulated from internal state instead of actual handshakes to HS split bus traffic.

Table 44-198. Summary of the Conditons of Handshakes[1]

Condition	Emulate TT Response
Start-Split: All asynchronous buffers full.	NAK
Start-Split: All periodic buffers full.	ERR
Start-Split: Success for start of Async. Transaction.	ACK
Start-Split: Start Periodic Transaction.	No Handshake (Ok)
Complete-Split: Failed to find transaction in queue.	Bus Time Out
Complete-Split: Transaction in Queue is Busy.	NYET
Complete-Split: Transaction in Queue is Complete.	[Actual Handshake from LS/FS device]

1. The un-shaded cells represent Start-Splits and the shaded cells represent Complete-Splits

44.6.4.1.5.3 Asynchronous Transaction Scheduling and Buffer Management

The following USB 2.0 specification items are implemented in the embedded Transaction Translator:

44.6.4.1.5.3.1 USB 2.0 - 11.17.3

- Sequencing is provided & a packet length estimator ensures no full-speed/low-speed packet babbles into SOF time.

44.6.4.1.5.3.2 USB 2.0 - 11.17.4

- Transaction tracking for 2 data pipes.

44.6.4.1.5.3.3 Periodic Transaction Scheduling and Buffer Management

The following USB 2.0 specification items are implemented in the embedded Transaction Translator:

44.6.4.1.5.3.4 USB 2.0 - 11.18.6.[1-2]

- Abort of pending start-splits
  - EOF (and not started in micro-frames 6)
  - Idle for more than 4 micro-frames
- Abort of pending complete-splits
  - EOF
  - Idle for more than 4 micro-frames

44.6.4.1.5.3.5 USB 2.0 - 11.18.[7-8]

- Transaction tracking for up to 16 data pipes.
- Complete-split transaction searching.

NOTE

There is no data schedule mechanism for these transactions other than the micro-frame pipeline. The embedded TT assumes the number of packets scheduled in a frame does not exceed the frame duration (1 ms) or else undefined behavior may result.

44.6.4.1.5.3.6 Multiple Transaction Translators

The maximum number of embedded Transaction Translators that is currently supported is one as indicated by the N_TT field in the Host Controller Structural Parameters (USB_HCSPARAMS) register.

44.6.4.2 Device Operation

The co-existence of a device operational controller within the host controller has little effect on EHCI compatibility for host operation except as noted in this section.

44.6.4.3 USB.USBMODE Register

Given that the dual-role controller is initialized in neither host nor device mode, the USB Device Mode (USB_USBMODE) register must be programmed for host operation before the EHCI host controller driver can begin EHCI host operations.

44.6.4.3.1 Non-Zero Fields the Register File

Some of the reserved fields and reserved addresses in the capability registers and operational register have use in device mode, the following must be adhered to:

- Write operations to all EHCI reserved fields (some of which are device fields) with the operation registers should always be written to zero. This is an EHCI requirement of the device controller driver that must be adhered to.
- Read operations by the host controller must properly mask EHCI reserved fields (some of which are device fields) because fields that are used exclusive for device are undefined in host mode .

44.6.4.3.2 SOF Interrupt

This SOF Interrupt used for device mode is shared as a free running 125us interrupt for host mode. EHCI does not specify this interrupt but it has been added for convenience and as a potential software time base. See USB Status Register (USB_USBSTS) and Interrupt Enable Register (USB_USBINTR) registers.

44.6.4.4 Embedded Design Interface

This is an Embedded USB Host Controller as defined by the EHCI specification and thus does not implement the PCI configuration registers.

44.6.4.4.1 Frame Adjust Register

Given that the optional PCI configuration registers are not included in this implementation, there is no corresponding bit level timing adjustments like is provided by the Frame Adjust register in the PCI configuration registers. Starts of micro-frames are timed precisely to 125 us using the transceiver clock as a reference clock. That is, a 60 Mhz transceiver clock for 8-bit physical interfaces & full-speed serial interfaces or 30 Mhz transceiver clock for 16-bit physical interfaces.

44.6.4.5 Miscellaneous variations from EHCI

44.6.4.5.1 Programmable Physical Interface Behaviour

This design supports multiple Physical interfaces which can operate in differing modes when the core is configured with software programmable Physical Interface Modes. Software programmability allows the selection of the Physical interface part during the board design phase instead of during the chip design phase. The control bits for selecting the Physical Interface operating mode have been added to the Port Status & Control (USB_PORTSC1) register providing a capability that is not defined by EHCI.

44.6.4.5.2 Discovery

44.6.4.5.2.1 Port Reset

The port connect methods specified by EHCI require setting the port reset bit in the Port Status & Control (USB_PORTSC1) register for a duration of 10ms. Due to the complexity required to support the attachment of devices that are not high speed there are counter already present in the design that can count the 10ms reset pulse to alleviate the requirement of the software to measure this duration. Therefore, the basic connection is then summarized as the following:

- [Port Change Interrupt] Port connect change occurs to notify the host controller driver that a device has attached.
- Software shall write a '1' to the reset the controller.
- After 10 ms, software shall write a '0' to the reset the controller.
  - This step, which is necessary in a standard EHCI design, may be omitted with this implementation. Should the EHCI host controller driver attempt to write a '0' to the reset bit while a reset is in progress the write will simple be ignored and the reset will continue until completion.
- [Port Change Interrupt] Port enable change occurs to notify the host controller that the device in now operational and at this point the port speed has been determined.

44.6.4.5.2.2 Port Speed Detection

After the port change interrupt indicates that a port is enabled, the EHCI stack should determine the port speed. Unlike the EHCI implementation which will re-assign the port owner for any device that does not connect at High-Speed, this host controller supports direct attach of non High-Speed devices. Therefore, the following differences are important regarding port speed detection:

- Port Owner is read-only and always reads 0.

- A 2-bit Port Speed indicator has been added to PORTSC to provide the current operating speed of the port to the host controller driver.
- A 1-bit High Speed indicator has been added to PORTSC to signify that the port is in High-Speed vs. Full/Low Speed - *This information is redundant with the 2-bit Port Speed indicator above.*

44.6.4.5.3 Port Test Mode

Port Test Control mode behaves fully as described in EHCI since the release of revision 3.2.1. In earlier product revisions, the test packet mode was not EHCI compatible. An alternate host controller driver procedure is no longer necessary or supported.

44.6.5 Device Data Structures

This section defines the interface data structures used to communicate control, status, and data between Device Controller Driver (DCD) Software and the Device Controller. The data structure definitions in this chapter support a 32-bit memory buffer address space. The interface consists of device Queue Heads and Transfer Descriptors.

NOTE

Software must ensure that no interface data structure reachable by the Device Controller spans a 4K-page boundary.

The data structures defined in the chapter are (from the device controller's perspective) a mix of read-only and read/ writeable fields. The device controller must preserve the read-only fields on all data structure writes.

The USB-HS OTG High-Speed USB On-The-Go core includes DCD Software called the USB 2.0 Device API. The Device API provides an easy to use Application Program Interface for developing device (peripheral) applications using the USB-HS OTG High-Speed USB On-The-Go core. The Device API incorporates and abstracts for the application developer all of the elements of the program interface.

Figure 44-165 shows the organization of the End Point Queue Head.

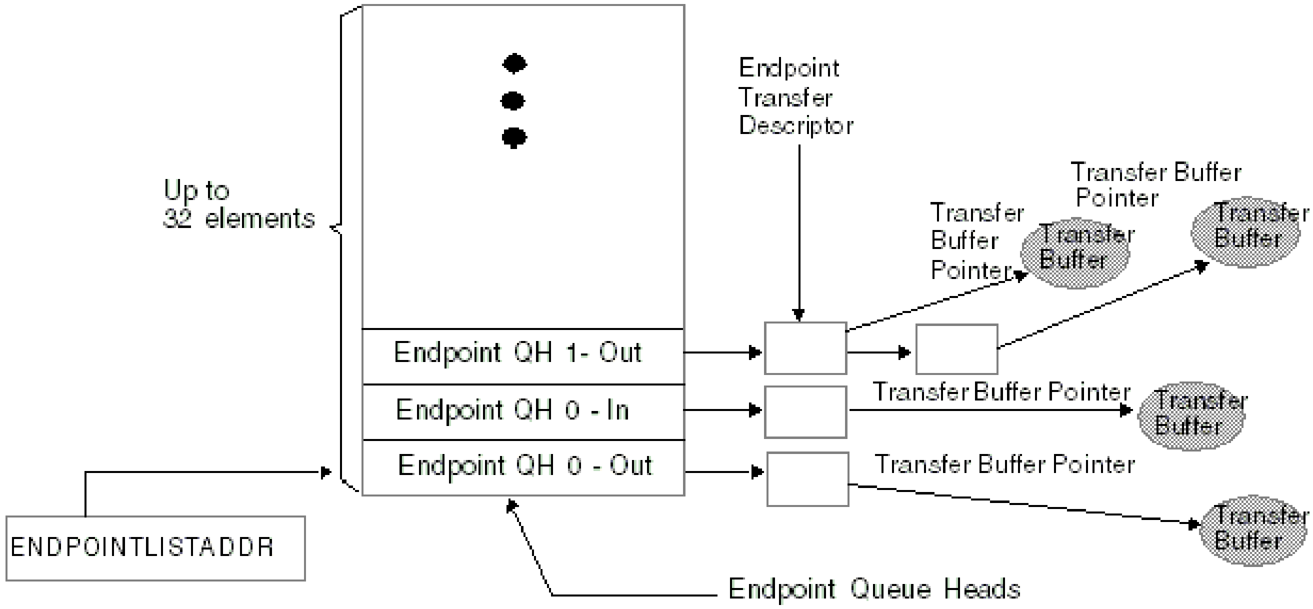


Figure 44-165. End Point Queue Head Organization

Device queue heads are arranged in an array in a continuous area of memory pointed to by the USB.ENDPOINTLISTADDR pointer. The even -numbered device queue heads in the list support receive endpoints (OUT/SETUP) and the odd-numbered queue heads in the list are used for transmit endpoints (IN/INTERRUPT). The device controller will index into this array based upon the endpoint number received from the USB bus. All information necessary to respond to transactions for all primed transfers is contained in this list so the Device Controller can readily respond to incoming requests without having to traverse a linked list.

NOTE

The Endpoint Queue Head List must be aligned to a 2k boundary.

44.6.5.1 Endpoint Queue Head (dQH)

The device Endpoint Queue Head (dQH) is where all transfers are managed. The dQH is a 48-byte data structure, but must be aligned on 64-byte boundaries. During priming of an endpoint, the dTD (device transfer descriptor) is copied into the overlay area of the dQH, which starts at the nextTD pointer DWord and continues through the end of the buffer pointers DWords. After a transfer is complete, the dTD status DWord is updated in the dTD pointed to by the currentTD pointer. While a packet is in progress, the overlay area of the dQH is used as a staging area for the dTD so that the Device Controller can access needed information with little minimal latency.

Table 44-199. Endpoint Queue Head (dQH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mult		zlt	0		Maximum Packet Length											ios	0														
Current dTD Pointer																											0				
Next dTD Pointer																											0				T
0	Total Bytes															ioc	0			Mult O		0		Status							
Buffer Pointer (Page 0)																				Current Offset											
Buffer Pointer (Page 1)																				-											
Buffer Pointer (Page 2)																				-											
Buffer Pointer (Page 3)																				-											
Buffer Pointer (Page 4)																				-											
-																															
Set-up Buffer Bytes 3…0																															
Set-up Buffer Bytes 7…4																															

	Device Controller Read/Write		Device Controller Read Only.

44.6.5.1.1 Endpoint Capabilities/Characteristics

This DWord specifies static information about the endpoint, in other words, this information does not change over the lifetime of the endpoint. Device Controller software should not attempt to modify this information while the corresponding endpoint is enabled.

Table 44-200 describes the endpoint capabilities.

Table 44-200. Endpoint Capabilities/Characteristics

Bit	Description
31-30	Mult. This field is used to indicate the number of packets executed per transaction description as given by the following: 00 - Execute N Transactions as demonstrated by the USB variable length packet protocol where N is computed using the Maximum Packet Length (dQH) and the Total Bytes field (dTD) 01 Execute 1 Transaction. 10 Execute 2 Transactions. 11 Execute 3 Transactions. **NOTE:** Non-ISO endpoints must set Mult="00". ISO endpoints must set Mult="01", "10", or "11" as needed.

Table continues on the next page...

Table 44-200. Endpoint Capabilities/Characteristics (continued)

29	Zero Length Termination Select. This bit is used to indicate when a zero length packet is used to terminate transfers where to total transfer length is a multiple . This bit is not relevant for Isochronous 0 - Enable zero length packet to terminate transfers equal to a multiple of the Maximum Packet Length. (default). 1 - Disable the zero length packet on transfers that are equal in length to a multiple Maximum Packet Length.
28-27	Reserved. These bit reserved for future use and should be set to zero.
26-16	Maximum Packet Length. This directly corresponds to the maximum packet size of the associated endpoint (wMaxPacketSize). The maximum value this field may contain is 0x400 (1024).
15	Interrupt On Setup (IOS). This bit is used on control type endpoints to indicate if USBINT is set in response to a setup being received.
14-0	Reserved. Bits reserved for future use and should be set to zero.

44.6.5.1.2 Transfer Overlay-Endpoint Queue Head

The seven DWords in the overlay area represent a transaction working space for the device controller. The general operational model is that the device controller can detect whether the overlay area contains a description of an active transfer. If it does not contain an active transfer, then it will not read the associated endpoint.

After an endpoint is readied, the dTD will be copied into this queue head overlay area by the device controller. Until a transfer is expired, software must not write the queue head overlay area or the associated transfer descriptor. When the transfer is complete, the device controller will write the results back to the original transfer descriptor and advance the queue.

See dTD for a description of the overlay fields.

44.6.5.1.3 Current dTD Pointer

The current dTD pointer is used by the device controller to locate the transfer in progress. This word is for Device Controller (hardware) use only and should not be modified by DCD software.

Table 44-201 describes the dTD Pointer.

Table 44-201. Next dTD Pointer

Bit	Description
31-5	Current dTD. This field is a pointer to the dTD that is represented in the transfer overlay area. This field will be modified by the Device Controller to next dTD pointer during endpoint priming or queue advance.
4-0	Reserved. Bit reserved for future use and should be set to zero.

44.6.5.1.4 Set-up Buffer

The set-up buffer is dedicated storage for the 8-byte data that follows a set-up PID.

NOTE

Each endpoint has a TX and an RX dQH associated with it, and only the RX queue head is used for receiving setup data packets.

Table 44-202 describes the Multiple Mode Control.

Table 44-202. Multiple Mode Control (HCCPARAMS)

DWord	Bits	Description
1	31-0	Setup Buffer 0. This buffer contains bytes 3 to 0 of an incoming setup buffer packet and is written by the device controller to be read by software.
2	31-0	Setup Buffer 1. This buffer contains bytes 7 to 4 of an incoming setup buffer packet and is written by the device controller to be read by software.

44.6.5.2 Endpoint Transfer Descriptor (dTD)

The dTD describes to the device controller the location and quantity of data to be sent/ received for given transfer. The DCD should not attempt to modify any field in an active dTD except the Next Like Pointer, which should only be modified as described in section Managing Transfers with Transfer Descriptors.

Table below shows the Endpoint Transfer Descriptor (dTD).

Table 44-203. Endpoint Transfer Descriptor (dTD)

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td colspan="27">Next Link Pointer</td><td colspan="4">0</td><td>T</td><td></td></tr>
<tr><td>0</td><td colspan="15">Total Bytes</td><td>ioc</td><td colspan="3">0</td><td colspan="2">MultO</td><td colspan="2">0</td><td colspan="8">Status</td><td></td></tr>
<tr><td colspan="20">Buffer Pointer (Page 0)</td><td colspan="12">Current Offset</td><td></td></tr>
<tr><td colspan="20">Buffer Pointer (Page 1)</td><td>0</td><td colspan="11">Frame Number</td><td></td></tr>
<tr><td colspan="20">Buffer Pointer (Page 2)</td><td colspan="12">-</td><td></td></tr>
<tr><td colspan="20">Buffer Pointer (Page 3)</td><td colspan="12">-</td><td></td></tr>
<tr><td colspan="20">Buffer Pointer (Page 4)</td><td colspan="12">Reserved</td><td></td></tr>
</table>

	Device Controller Read/Write		Device Controller Read Only.

Table 44-204 describes the dTD Pointer.

Table 44-204. Next dTD Pointer

Bit	Description
31-5	Next Transfer Element Pointer. This field contains the physical memory address of the next dTD to be processed. The field corresponds to memory address signals [31:5], respectively.
4-1	Reserved. Bits reserved for future use and should be set to zero.
0	Terminate (T). 1=pointer is invalid. 0=Pointer is valid (points to a valid Transfer Element Descriptor). This bit indicates to the Device Controller that there are no more valid entries in the queue.

Table 44-205 describes the dTD Token.

Table 44-205. dTD Token

Bit	Description
31	Reserved. Bit reserved for future use and should be set to zero.
30-16	Total Bytes. This field specifies the total number of bytes to be moved with this transfer descriptor. This field is decremented by the number of bytes actually moved during the transaction and only on the successful completion of the transaction. The maximum value software may store in the field is 5*4K(5000H). This is the maximum number of bytes 5 page pointers can access. Although it is possible to create a transfer up to 20K this assumes the 1st offset into the first page is 0. When the offset cannot be predetermined, crossing past the 5th page can be guaranteed by limiting the total bytes to 16K**. Therefore, the maximum recommended transfer is 16K(4000H). If the value of the field is zero when the host controller fetches this transfer descriptor (and the active bit is set), the device controller executes a zero-length transaction and retires the transfer descriptor. It is not a requirement for IN transfers that Total Bytes To Transfer be an even multiple of *Maximum Packet Length*. If software builds such a transfer descriptor for an IN transfer, the last transaction will always be less that *Maximum Packet Length*.
15	Interrupt On Complete (IOC). This bit is used to indicate if USBINT is to be set in response to device controller being finished with this dTD.
14-12	Reserved. Bits reserved for future use and should be set to zero.
11-10	Multiplier Override (MultO). This field can be used for transmit ISO's (ie. ISO-IN) to override the multiplier in the QH. This field must be zero for all packet types that are not transmit-ISO. Example: if QH.multiplier = 3; Maximum packet size = 8; Total Bytes = 15; MultiO = 0 [default] Three packets are sent: {Data2(8); Data1(7); Data0(0)} if QH.multiplier = 3; Maximum packet size = 8; Total Bytes = 15; MultiO = 2 Two packets are sent: {Data1(8); Data0(7)} For maximal efficiency, software should compute MultO = greatest integer of (Total Bytes / Max. Packet Size) except for the case when Total Bytes = 0; then MultO should be 1. Note: Non-ISO and Non-TX endpoints must set MultO="00".
9-8	Reserved. Bits reserved for future use and should be set to zero.
7-0	Status. This field is used by the Device Controller to communicate individual command execution states back to the Device Controller software. This field contains the status of the last transaction performed on this qTD. The bit encodings are: Bit Status Field Description 7 Active. 6 Halted. 5 Data Buffer Error. 3 Transaction Error. 4,2,0 Reserved.

Table 44-206 describes the dTD Buffer Page Pointer List.

Table 44-206. dTD Buffer Page Pointer List

Bit	Description
31-12	Buffer Pointer. Selects the page offset in memory for the packet buffer. Non virtual memory systems will typically set the buffer pointers to a series of incrementing integers.
0,11-0	Current Offset. Offset into the 4kb buffer where the packet is to begin.
1,10-0	Frame Number. Written by the device controller to indicate the frame number in which a packet finishes. This is typically be used to correlate relative completion times of packets on an ISO endpoint.

44.6.6 Device Operational Model

The function of the device operation is to transfer a request in the memory image to and from the Universal Serial Bus. Using a set of linked list transfer descriptors, pointed to by a queue head, the device controller will perform the data transfers. The following sections explain the use of the device controller from the device controller driver (DCD) point-of-view and further describe how specific USB bus events relate to status changes in the device controller programmer's interface.

44.6.6.1 Device Controller Initialization

After hardware reset, the device is disabled until the Run/Stop bit is set to a '1'. In the disabled state, the pull-up on the USB D+ is not active which prevents an attach event from occurring. At a minimum, it is necessary to have the queue heads setup for endpoint zero before the device attach occurs. Shortly after the device is enabled, a USB reset will occur followed by setup packet arriving at endpoint 0. A Queue head must be prepared so that the device controller can store the incoming setup packet.

In order to initialize a device, the software should perform the following steps:

1. Set Controller Mode in the USBx_USBMODE register to device mode.
2. Allocate and Initialize device queue heads in system memory.

NOTE

Transitioning from host mode to device mode requires a device controller reset before modifying USBx_USBMODE.

- Minimum: Initialize device queue heads 0 Tx & 0 Rx.
- For information on device queue heads, refer to section Device Data Structures.

NOTE

All device queue heads associated with control endpoints must be initialized before the control endpoint is enabled. Non-Control device queue heads before the endpoint can be used.

3. Configure USB.ENDPOINTLISTADDR Pointer.
    - For additional information on USB.ENDPOINTLISTADDR, refer to the register table.
4. Enable the microprocessor interrupt associated with the USB-HS OTG High-Speed USB On-The-Go core.
    - Recommended: enable all device interrupts including: USBINT, USBERRINT, USB Reset Received, DCSuspend.
    - For a list of available interrupts refer to the Interrupt Enable Register (USB_USBINTR) and the USB Status Register (USB_USBSTS) register tables.
5. Set Run/Stop bit to Run Mode.
    - After the Run bit is set, a device reset will occur. The DCD must monitor the reset event and adjust the software state as described in the Bus Reset section of the following Port State and Control section below.

NOTE

Endpoint 0 is designed as a control endpoint only and does not need to be configured using ENDPTCTRL0 register.

It is also not necessary to initially prime Endpoint 0 because the first packet received will always be a setup packet. The contents of the first setup packet will require a response in accordance with USB device framework (Chapter 9) command set.

44.6.6.2 Port State and Control

From a chip or system reset, the device controller enters the *powered* state. A transition from the *powered* state to the *attach* state occurs when the Run/Stop bit is set to a '1'. After receiving a reset on the bus, the port will enter the *defaultFS or defaultHS* state in accordance with the reset protocol described in Appendix C.2 of the USB Specification Rev. 2.0. The following state diagram depicts the state of a USB 2.0 device.

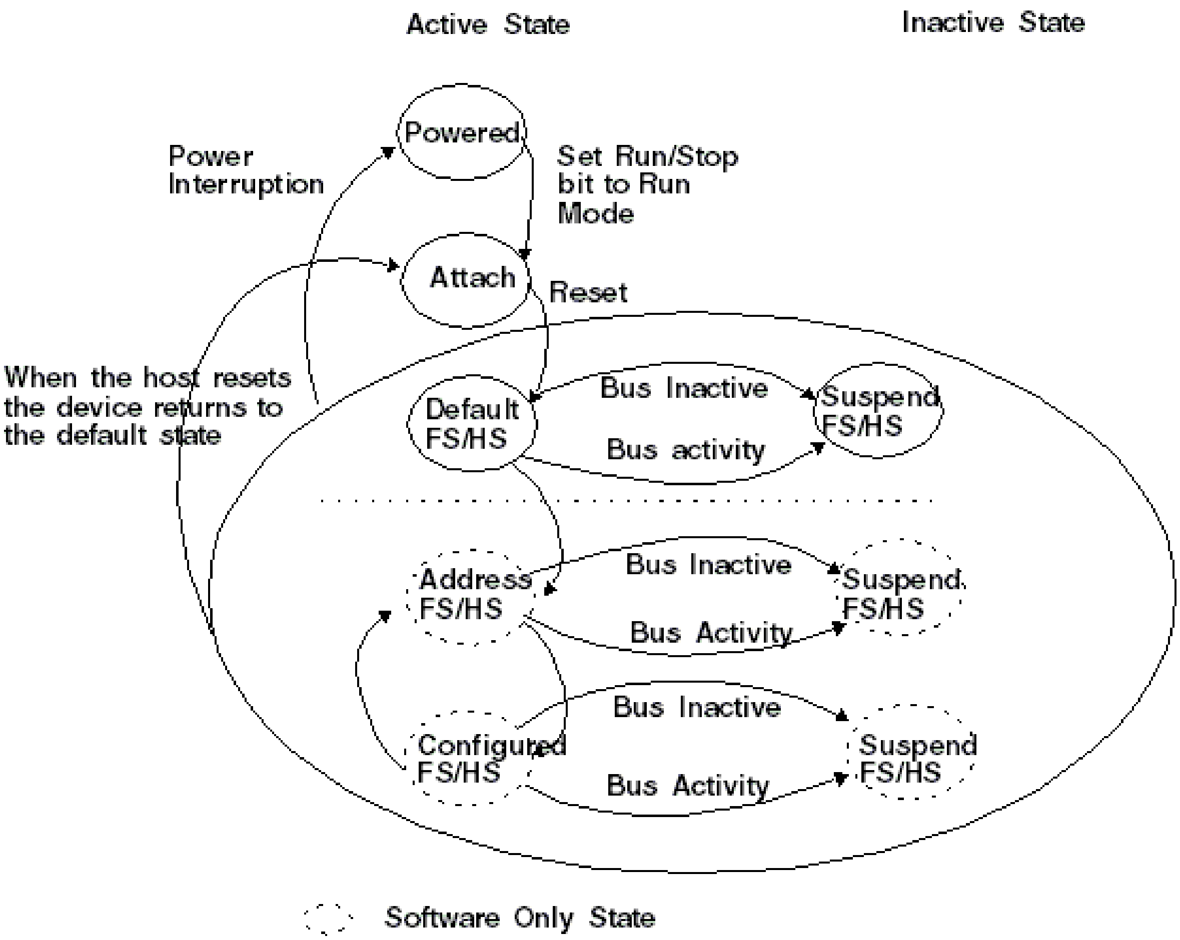


Figure 44-166. Device State Diagram

States *powered, attach, defaultFS/HS, suspendFS/HS* are implemented in the device controller and are communicated to the DCD using the following status bits:

Table 44-207 describes the Device Controller State Information Bits.

Table 44-207. Device Controller State Information Bits

Bit	Register
DCSuspend	USB Status Register (USB_USBSTS)
USB Reset Received	USBSTS
Port Change Detect	USBSTS
High-Speed Port	Port Status & Control (USB_PORTSC1)

It is the responsibility of the DCD to maintain a state variable to differentiate between the *DefaultFS/HS* state and the *Address/Configured* states. Change of state from *Default* to *Address* and the *Configured* states is part of the enumeration process described in the device framework section of the USB 2.0 Specification.

As a result of entering the *Address* state, the device address register (DEVICEADDR) must be programmed by the DCD.

Entry into the *Configured* indicates that all endpoints to be used in the operation of the device have been properly initialized by programming the ENDPTCTRLx registers and initializing the associated queue heads.

44.6.6.2.1 Bus Reset

A bus reset is used by the host to initialize downstream devices. When a bus reset is detected, the device controller will renegotiate its attachment speed, reset the device address to 0, and notify the DCD by interrupt (assuming the USB Reset Interrupt Enable is set). After a reset is received, all endpoints (except endpoint 0) are disabled and any primed transactions will be cancelled by the device controller. The concept of priming will be clarified below, but the DCD must perform the following tasks when a reset is received:

Clear all setup token semaphores by reading the Endpoint Status (USB_ENDPTSTAT) register and writing the same value back to the Endpoint Status (USB_ENDPTSTAT) register.

Clear all the endpoint complete status bits by reading the Endpoint Complete (USB_ENDPTCOMPLETE) register and writing the same value back to the Endpoint Complete (USB_ENDPTCOMPLETE) register.

Cancel all primed status by waiting until all bits in the Endpoint Initialization (USB_ENDPTPRIME) are 0 and then writing 0xFFFFFFFF to Endpoint De-Initialize (USB_ENDPTFLUSH).

Read the reset bit in the Port Status & Control (USB_PORTSC1) register and make sure that it is still active. A USB reset will occur for a minimum of 3 ms and the DCD must reach this point in the reset cleanup before end of the reset occurs, otherwise a hardware reset of the device controller is recommended (rare.)

- A hardware reset can be performed by writing a one to the device controller reset bit in the USBCMD reset. Note: a hardware reset will cause the device to detach from the bus by clearing the Run/Stop bit. Thus, the DCD must completely re-initialize the device controller after a hardware reset.

Free all allocated dTDs because they will no longer be executed by the device controller. If this is the first time the DCD is processing a USB reset event, then it is likely that no dTDs have been allocated.

At this time, the DCD may release control back to the OS because no further changes to the device controller are permitted until a Port Change Detect is indicated.

After a Port Change Detect, the device has reached the default state and the DCD can read the Port Status & Control (USB_PORTSC1) to determine if the device is operating in FS or HS mode. At this time, the device controller has reached normal operating mode and DCD can begin enumeration according to the USB Chapter 9 - Device Framework.

NOTE

The device DCD may use the FS/HS mode information to determine the bandwidth mode of the device

In some applications, it may not be possible to enable one or more pipes while in FS mode. *Beyond the data rate issue, there is no difference in DCD operation between FS and HS modes.*

44.6.6.2.2 Suspend/Resume

44.6.6.2.2.1 Suspend

Suspend Description

In order to conserve power, USB devices automatically enter the suspended state when the device has observed no bus traffic for a specified period. When suspended, the USB device maintains any internal status, including its address and configuration. Attached devices must be prepared to suspend at any time they are powered, regardless of if they have been assigned a non-default address, are configured, or neither. Bus activity may cease due to the host entering a suspend mode of its own. In addition, a USB device shall also enter the suspended state when the hub port it is attached to is disabled.

A USB device exits suspend mode when there is bus activity. A USB device may also request the host to exit suspend mode or selective suspend by using electrical signaling to indicate remote wakeup. The ability of a device to signal remote wakeup is optional. If the USB device is capable of remote wakeup signaling, the device must support the ability of the host to enable and disable this capability. When the device is reset, remote wakeup signaling must be disabled.

Suspend Operational Model

The device controller moves into the suspend state when suspend signaling is detected or activity is missing on the upstream port for more than a specific period. After the device controller enters the suspend state, the DCD is notified by an interrupt (assuming *DC Suspend Interrupt* is enabled). When the *DCSuspend* bit in the Port Status & Control (USB_PORTSC1) is set to a '1', the device controller is suspended.

DCD response when the device controller is suspended is application specific and may involve switching to low power operation.

Information on the bus power limits in suspend state can be found in USB 2.0 specification.

NOTE

Review system level clocking issues defined in section (Ref: Signals-Clocking) for the clocking requirements of a suspended device controller.

44.6.6.2.2.2 Resume

If the device controller is suspended, its operation is resumed when any non-idle signaling is received on its upstream facing port. In addition, the device can signal the system to resume operation by forcing resume signaling to the upstream port. Resume signaling is sent upstream by writing a '1' to the Resume bit in the in the Port Status & Control (USB_PORTSC1) while the device is in suspend state. Sending resume signal to an upstream port should cause the host to issue resume signaling and bring the suspended bus segment (one more devices) back to the active condition.

NOTE

Before resume signaling can be used, the host must enable it by using the Set Feature command defined in device framework (chapter 9) of the USB 2.0 Specification.

44.6.6.2.3 Managing Endpoints

The USB 2.0 specification defines an endpoint, also called a device endpoint or an address endpoint as a uniquely addressable portion of a USB device that can source or sink data in a communications channel between the host and the device. The endpoint address is specified by the combination of the endpoint number and the endpoint direction.

The channel between the host and an endpoint at a specific device represents a data pipe. Endpoint 0 for a device is always a *control* type data channel used for device discovery and enumeration. Other types of endpoints support by USB include *bulk, interrupt*, and *isochronous*. Each endpoint type has specific behavior related to packet response and error handling. More detail on endpoint operation can be found in the USB 2.0 specification.

Each endpoint direction is essentially independent and can be configured with differing behavior in each direction. For example, the DCD can configure endpoint 1-IN to be a bulk endpoint and endpoint 1-OUT to be an isochronous endpoint. This helps to conserve the total number of endpoints required for device operation. The only exception is that

control endpoints must use both directions on a single endpoint number to function as a control endpoint. Endpoint 0 is, for example, is always a control endpoint and uses the pair of directions.

Each endpoint direction requires a *queue head* allocated in memory. If the maximum of 16 endpoint numbers, one for each endpoint direction are being used by the device controller, then 32 *queue heads* are required. The operation of an endpoint and use of *queue heads* are described later in this document.

44.6.6.2.4 Endpoint Initialization

After hardware reset, all endpoints except endpoint zero are uninitialized and disabled. The DCD must configure and enable each endpoint by writing to configuration bit in the ENDPTCTRLx register. Each 32-bit ENDPTCTRLx is split into an upper and lower half. The lower half of ENDPTCTRLx is used to configure the receive or OUT endpoint and the upper half is likewise used to configure the corresponding transmit or IN endpoint. Control endpoints must be configured the same in both the upper and lower half of the ENDPTCTRLx register otherwise the behavior is undefined. The following table shows how to construct a configuration word for endpoint initialization. Table 44-208 shows the fields and values for the Device Controller Endpoint initialization.

Table 44-208. Device Controller Endpoint Initialization

Field	Value
Data Toggle Reset	1
Data Toggle Inhibit	0
Endpoint Type	00 Control 01 Isochronous 10 Bulk 11 Interrupt
Endpoint Stall	0

44.6.6.2.5 Stalling

There are two occasions where the device controller may need to return to the host a STALL

The first occasion is the functional stall, which is a condition set by the DCD as described in the USB 2.0 device framework. A functional stall is only used on non-control endpoints and can be enabled in the device controller by setting the endpoint stall bit in the ENDPTCTRLx register associated with the given endpoint and the given direction. In

a functional stall condition, the device controller will continue to return STALL responses to all transactions occurring on the respective endpoint and direction until the endpoint stall bit is cleared by the DCD.

A protocol stall, unlike a function stall, is used on control endpoints is automatically cleared by the device controller at the start of a new control transaction (setup phase). When enabling a protocol stall, the DCD should enable the stall bits (both directions) as a pair. A single write to the ENDPTCTRLx register can ensure that both stall bits are set at the same instant.

NOTE

Any write to the ENDPTCTRLx register during operational mode must preserve the endpoint type field (that is, perform a read-modify-write).

Table 44-209 shows the response matrix for the Device Controller Stall.

Table 44-209. Device Controller Stall Response Matrix

USB Packet	Endpoint Stall Bit.	Effect on STALL bit.	USB Response
SETUP packet received by a non-control endpoint.	N/A	None.	STALL
IN/OUT/PING packet received by a non-control endpoint.	'1'	None.	STALL
IN/OUT/PING packet received by a non-control endpoint.	'0'	None.	ACK/ NAK/ NYET
SETUP packet received by a control endpoint.	N/A	Cleared	ACK
IN/OUT/PING packet received by a control endpoint	'1'	None	STALL
IN/OUT/PING packet received by a control endpoint.	'0'	None.	ACK/ NAK/ NYET

44.6.6.2.6 Data Toggle

Data toggle is a mechanism to maintain data coherency between host and device for any given data pipe. For more information on data toggle, refer to the USB 2.0 specification.

44.6.6.2.6.1 Data Toggle Reset

The DCD may reset the data toggle state bit and cause the data toggle sequence to reset in the device controller by writing a '1' to the data toggle reset bit in the ENDPTCTRLx register. This should only be necessary when configuring/initializing an endpoint or returning from a STALL condition.

44.6.6.2.6.2 Data Toggle Inhibit

NOTE

This feature is for test purposes only and should never be used during normal device controller operation.

Setting the *data toggle Inhibit bit* active ('1') causes the device controller to ignore the data toggle pattern that is normally sent and accept all incoming data packets regardless of the data toggle state.

In normal operation, the device controller checks the DATA0/DATA1 bit against the data toggle to determine if the packet is valid. If Data PID does not match the data toggle state bit maintained by the device controller for that endpoint, the Data toggle is considered not valid. If the data toggle is not valid, the device controller assumes the packet was already received and discards the packet (not reporting it to the DCD). To prevent the host controller from re-sending the same packet, the device controller will respond to the error packet by acknowledging it with either an ACK or NYET response.

44.6.6.2.6.3 Priming Transmit Endpoints

Priming a transmit endpoint will cause the device controller to fetch the device transfer descriptor (dTD) for the transaction pointed to by the device queue head (dQH). After the dTD is fetched, it will be stored in the dQH until the device controller completes the transfer described by the dTD. Storing the dTD in the dQH allows the device controller to fetch the operating context needed to handle a request from the host without the need to follow the linked list, starting at the dQH when the host request is received.

After the device has loaded the dTD, the leading data in the packet is stored in a FIFO in the device controller. This FIFO is split into virtual channels so that the leading data can be stored for any endpoint up to the maximum number of endpoints configured at device synthesis time.

After a priming request is complete, an endpoint state of primed is indicated in the USB.ENDPTSTATUS register. For a primed transmit endpoint, the device controller can respond to an IN request from the host and meet the stringent bus turnaround time of High Speed USB.

Because only the leading data is stored in the device controller FIFO, it is necessary for the device controller to begin filling in behind leading data after the transaction starts. The FIFO must be sized to account for the maximum latency that can be incurred by the system memory bus. More information about FIFO sizing is presented in section .

44.6.6.2.6.4 Priming Receive Endpoints

Priming receive endpoints is identical to priming of transmit endpoints from the point of view of the DCD. At the device controller the major difference in the operational model is that there is no data movement of the leading packet data simply because the data is to be received from the host.

Note as part of the architecture, the FIFO for the receive endpoints is not partitioned into multiple channels like the transmit FIFO. Thus, the size of the RX FIFO does not scale with the number of endpoints.

44.6.6.3 Operational Model For Packet Transfers

All transactions on the USB bus are initiated by the host and in turn, the device must respond to any request from the host within the turnaround time stated in the USB 2.0 Specification. At USB 1.1 Full or Low Speed rates, this turnaround time was significant and the USB 1.1 device controllers were architected so that the device controller could access main memory or interrupt a host protocol processor in order to respond to the USB 1.1 transaction. The architecture of the USB 2.0 device controller must be different because same methods will not meet USB 2.0 High-speed turnaround time requirements by simply increasing clock rate.

A USB host will send requests to the device controller in an order that can not be precisely predicted as a single pipeline, so it is not possible to prepare a single packet for the device controller to execute. However, the order of packet requests is predictable when the endpoint number and direction is considered. For example, if endpoint 3 (transmit direction) is configured as a bulk pipe, then we can expect the host will send IN requests to that endpoint. This device controller is architected in such a way that it can prepare packets for each endpoint/direction in anticipation of the host request. The process of preparing the device controller to send or receive data in response to host initiated transaction on the bus is referred to as "priming" the endpoint. This term will be used throughout the following documentation to describe the device controller operation so the DCD can be architected properly use priming. Further, note that the term "flushing" is used to describe the action of clearing a packet that was queued for execution.

44.6.6.3.1 Interrupt/Bulk Endpoint Operational Model

The behaviors of the device controller for interrupt and bulk endpoints are identical. All valid IN and OUT transactions to bulk pipes will handshake with a NAK unless the endpoint had been primed. Once the endpoint has been primed, data delivery will commence.

A dTD will be retired by the device controller when the packets described in the transfer descriptor have been completed. Each dTD describes N packets to be transferred according to the USB Variable Length transfer protocol. The formula and table on the following page describe how the device controller computes the number and length of the packets to be sent/received by the USB vary according to the total number of bytes and maximum packet length.

With Zero Length Termination (ZLT) = 0

N = INT(Number Of Bytes/Max. Packet Length) + 1

With Zero Length Termination (ZLT) = 1

N = MAXINT(Number Of Bytes/Max. Packet Length)

Table 44-210. Variable Length Transfer Protocol Example (ZLT = 0)

Bytes (dTD)	Max. Packet Length (dQH)	N	P1	P2	P3
511	256	2	256	255	
512	256	3	256	256	0
512	512	2	512	0	

Table 44-211. Variable Length Transfer Protocol Example (ZLT = 1)

Bytes (dTD)	Max. Packet Length (dQH)	N	P1	P2	P3
511	256	2	256	255	
512	256	2	256	256	
512	512	1	512		

NOTE

The MULT field in the dQH must be set to "00" for bulk, interrupt, and control endpoints.

TX-dTD is complete when:

- All packets described dTD were successfully transmitted. *** Total bytes in dTD will equal zero when this occurs.

RX-dTD is complete when:

- All packets described in dTD were successfully received. *** Total bytes in dTD will equal zero when this occurs.
- A short packet (number of bytes < maximum packet length) was received. *** This is a successful transfer completion; DCD must check Total Bytes in dTD to determine the number of bytes that are remaining. From the total bytes remaining in the dTD, the DCD can compute the actual bytes received.
- A long packet was received (number of bytes > maximum packet size) OR (total bytes received > total bytes specified). *** This is an error condition. The device controller will discard the remaining packet, and set the Buffer Error bit in the dTD. In addition, the endpoint will be flushed and the USBERR interrupt will become active.

On the successful completion of the packet(s) described by the dTD, the active bit in the dTD will be cleared and the next pointer will be followed when the Terminate bit is clear. When the Terminate bit is set, the device controller will flush the endpoint/direction and cease operations for that endpoint/direction.

On the unsuccessful completion of a packet (see long packet above), the dQH will be left pointing to the dTD that was in error. In order to recover from this error condition, the DCD must properly reinitialize the dQH by clearing the active bit and update the nextTD pointer before attempting to re-prime the endpoint.

NOTE

All packet level errors such as a missing handshake or CRC error will be retried automatically by the device controller.

There is no required interaction with the DCD for handling such errors.

44.6.6.3.1.1 Interrupt/Bulk Endpoint Bus Response Matrix

Table 44-212 shows the response matrix for Interrput/Bulk Endpoint Bus.

Table 44-212. Interrupt/Bulk Endpoint Bus Response Matrix

	Stall	Not Primed	Primed	Underflow	Overflow
Setup	Ignore	Ignore	Ignore	N/A	N/A
In	STALL	NAK	Transmit	BS Error	N/A
Out	STALL	NAK	Receive + NYET/ACK	N/A	NAK
Ping	STALL	NAK	ACK	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

NOTE

BS Error = Force Bit Stuff Error

NYET/ACK - NYET unless the Transfer Descriptor has packets remaining according to the USB variable length protocol then ACK.

SYSERR - System error should never occur when the latency FIFOs are correctly sized and the DCD is responsive.

44.6.6.3.2 Control Endpoint Operation Model

44.6.6.3.2.1 Setup Phase

All requests to a control endpoint begin with a setup phase followed by an optional data phase and a required status phase. The device controller will always accept the setup phase unless the setup lockout is engaged.

The setup lockout will engage so that future setup packets are ignored. Lockout of setup packets ensures that while software is reading the setup packet stored in the queue head, that data is not written as it is being read potentially causing an invalid setup packet.

In hardware versions 2.3 and later, the setup lockout mechanism can be disabled and a new tripwire type semaphore will ensure that the setup packet payload is extracted from the queue head without being corrupted be an incoming setup packet. This is the preferred behavior because ignoring repeated setup packets due to long software interrupt latency would be a compliance issue.

Setup Packet Handling (Pre-2.3 hardware)

- After receiving an interrupt and inspecting USB Device Mode (USB_USBMODE) to determine that a setup packet was received on a particular pipe:

1. Duplicate contents of dQH.SsetupBuffer into local software byte array.
2. Write '1' to clear corresponding Endpoint Setup Status (USB_ENDPTSETUPSTAT) bit and thereby disabling Setup Lockout. (That is, the Setup Lockout activates as soon as a setup arrives. By writing to the Endpoint Setup Status (USB_ENDPTSETUPSTAT), the device controller will accept new setup packets.)
3. Process setup packet using local software byte array copy and execute status/ handshake phases.

NOTE

After receiving a new setup packet the status and/or handshake phases may still be pending from a previous control sequence. These should be flushed & deallocated before linking a new status and/or handshake dTD for the most recent setup packet.

NOTE

To limit the exposure of setup packets to the setup lockout mechanism (if used), the DCD should designate the priority of responding to setup packets above responding to other packet completions.

Setup Packet Handling (2.3 hardware and later)

- Disable Setup Lockout by writing 1 to Setup Lockout Mode (SLOM) in USB Device Mode (USB_USBMODE). (once at initialization). Setup lockout is not necessary when using the tripwire as described below.

NOTE

Leaving the Setup Lockout Mode As 0 will result in pre-2.3 hardware behavior.

- After receiving an interrupt and inspecting Endpoint Setup Status (USB_ENDPTSETUPSTAT) to determine that a setup packet was received on a particular pipe:
  a. Write 1 to clear corresponding bit Endpoint Setup Status (USB_ENDPTSETUPSTAT).
  b. Write 1 to Setup Tripwire (SUTW) in USB Command Register (USB_USBCMD) register.
  c. Duplicate contents of dQH.SetupBuffer into local software byte array.
  d. Read Setup TripWire (SUTW) in USB Command Register (USB_USBCMD) register. (if set - continue; if cleared - goto 2)
  e. Write 0 to clear Setup Tripwire (SUTW) in USB Command Register (USB_USBCMD) register.
  f. Process setup packet using local software byte array copy and execute status/ handshake phases.

NOTE

After receiving a new setup packet the status and/or handshake phases may still be pending from a previous control sequence. These should be flushed & deallocated before linking a new status and/or handshake dTD for the most recent setup packet.

44.6.6.3.2.2 Data Phase

Following the setup phase, the DCD must create a device transfer descriptor for the data phase and prime the transfer.

After priming the packet, the DCD must verify a new setup packet has not been received by reading the USB.ENDPTSETUPSTAT register immediately verifying that the prime had completed. A prime will complete when the associated bit in the Endpoint Initialization (USB_ENDPTPRIME) register is zero and the associated bit in the Endpoint Status (USB_ENDPTSTAT) register is a one. If a prime fails, ie. The Endpoint Initialization (USB_ENDPTPRIME) bit goes to zero and the Endpoint Status (USB_ENDPTSTAT) bit is not set, then the prime has failed. This can only be due to improper setup of the dQH, dTD or a setup arriving during the prime operation. If a new setup packet is indicated after the ENDPTPRIME bit is cleared, then the transfer descriptor can be freed and the DCD must reinterpret the setup packet.

Should a setup arrive after the data stage is primed, the device controller will automatically clear the prime status (Endpoint Status (USB_ENDPTSTAT)) to enforce data coherency with the setup packet.

NOTE

- The MULT field in the dQH must be set to "00" for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is the same as bulk packets described previously.

44.6.6.3.2.3 Status Phase

Similar to the data phase, the DCD must create a transfer descriptor (with byte length equal zero) and prime the endpoint for the status phase. The DCD must also perform the same checks of the USB.ENDPTSETUPSTAT as described above in the data phase.

NOTE

- The MULT field in the dQH must be set to 00 for bulk, interrupt, and control endpoints.
- Error handling of data phase packets is the same as bulk packets described previously.

44.6.6.3.2.4 Control Endpoint Bus Response Matrix

Shown in the following table is the device controller response to packets on a control endpoint according to the device controller state.

Table 44-213 shows the response matrix for the Control Endpoint Bus.

Table 44-213. Control Endpoint Bus Response Matrix

Token Type	Endpoint State					Setup Lockout
	Stall	Not Primed	Primed	Underflow	Overflow	

Table continues on the next page...

Table 44-213. Control Endpoint Bus Response Matrix (continued)

Setup	ACK	ACK	ACK	N/A	SYSERR	
In	STALL	NAK	Transmit	BS Error	N/A	N/A
Out	STALL	NAK	Receive + NYET/ ACK	N/A	NAK	N/A
Ping	STALL	NAK	ACK	N/A	N/A	N/A
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore	Ignore

BS Error = Force Bit Stuff Error

NYET/ACK - NYET unless the Transfer Descriptor has packets remaining according to the USB variable length protocol then ACK.

SYSERR - System error should never occur when the latency FIFOs are correctly sized and the DCD is responsive.

44.6.6.3.3 Isochronous Endpoint Operational Model

Isochronous endpoints are used for real-time scheduled delivery of data and their operational model is significantly different than the host throttled Bulk, Interrupt, and Control data pipes. Real time delivery by the device controller will is accomplished by the following:

- Exactly MULT Packets per (micro)Frame are transmitted/received. Note: MULT is a 2-bit field in the device Queue Head. The variable length packet protocol is not used on isochronous endpoints.
- NAK responses are not used. Instead, zero length packets and sent in response to an IN request to an unprimed endpoints. For unprimed RX endpoints, the response to an OUT transaction is to ignore the packet within the device controller.
- Prime requests always schedule the transfer described in the dTD for the next (micro)frame. If the ISO-dTD is still active after that frame, then the ISO-dTD will be held ready until executed or canceled by the DCD.

An EHCI compatible host controller uses the periodic frame list to schedule data exchanges to Isochronous endpoints. The operational model for device mode does not use such a data structure. Instead, the same dTD used for Control/Bulk/Interrupt endpoints is also used for isochronous endpoints. The difference is in the handling of the dTD.

The first difference between bulk and ISO-endpoints is that priming an ISO-endpoint is a delayed operation such that an endpoint will become primed only after a SOF is received. After the DCD writes the prime bit, the prime bit will be cleared as usual to indicate to software that the device controller completed a priming the dTD for transfer. Internal to

the design, the device controller hardware masks that prime start until the next frame boundary. This behavior is hidden from the DCD but occurs so that the device controller can match the dTD to a specific (micro)frame.

Another difference with isochronous endpoints is that the transaction must wholly complete in a (micro)frame. Once an ISO transaction is started in a (micro)frame it will retire the corresponding dTD when MULT transactions occur or the device controller finds a fulfillment condition.

The transaction error bit set in the status field indicates a fulfillment error condition. When a fulfillment error occurs, the frame after the transfer failed to complete wholly, the device controller will force retire the ISO-dTD and move to the next ISO-dTD.

It is important to note that fulfillment errors are only caused due to partially completed packets. If no activity occurs to a primed ISO-dTD, the transaction will stay primed indefinitely. This means it is up to software discard transmit ISO-dTDs that pile up from a failure of the host to move the data.

Finally, the last difference with ISO packets is in the data level error handling. When a CRC error occurs on a received packet, the packet is not retried similar to bulk and control endpoints. Instead, the CRC is noted by setting the *Transaction Error* bit and the data is stored as usual for the application software to sort out.

- TX Packet Retired
    - MULT counter reaches zero.
    - Fulfillment Error [*Transaction Error* bit is set]
        - # Packets Occurred > 0 AND # Packets Occurred < MULT

NOTE

For TX-ISO, MULT Counter can be loaded with a lesser value in the dTD Multiplier Override field in hardware versions 2.3 and later. If the Multiplier Override is zero, the MULT Counter is initialized to the Multiplier in the QH.

- RX Packet Retired:
    - MULT counter reaches zero.
    - Non-MDATA Data PID is received**
        - ** Exit criteria only valid in hardware version 2.3 or later. Previous to hardware version 2.3, any PID sequence that did not match the MULT field exactly would be flagged as a transaction error due to PID mismatch or fulfillment error.
    - Overflow Error:
        - Packet received is > maximum packet length. [*Buffer Error* bit is set]
        - Packet received exceeds total bytes allocated in dTD. [*Buffer Error* bit is set]

- Fulfillment Error [*Transaction Error* bit is set]
  - # Packets Occurred > 0 AND # Packets Occurred < MULT
- CRC Error [*Transaction Error* bit is set

NOTE

For ISO, when a dTD is retired, the next dTD is primed for the next frame. For continuous (micro)frame to (micro)frame operation the DCD should ensure that the dTD linked-list is out ahead of the device controller by at least two (micro)frames.

44.6.6.3.3.1 Isochronous Pipe Synchronization

When it is necessary to synchronize an isochroous data pipe to the host, the (micro)frame number (USB.FRINDEX register) can be used as a marker. To cause a packet transfer to occur at a specific (micro)frame number [N], the DCD should interrupt on SOF during frame N-1. When the USB.FRINDEX=N-1, the DCD must write the prime bit. The device controller will prime the isochronous endpoint in (micro)frame N-1 so that the device controller will execute delivery during (micro)frame N.

NOTE

Priming an endpoint towards the end of (micro)frame N-1 will not guarantee delivery in (micro)frame N. The delivery may actually occur in (micro)frame N+1 if device controller does not have enough time to complete the prime before the SOF for packet N is received.

44.6.6.3.3.2 Isochronous Endpoint Bus Response Matrix

Table 44-214 shows the response matrix for the Isochronous Endpoint Bus.

Table 44-214. Isochronous Endpoint Bus Response Matrix

	Stall	Not Primed	Primed	Underflow	Overflow
Setup	STALL	STALL	STALL	N/A	N/A
In	NULL Packet	NULL Packet	Transmit	BS Error	N/A
Out	Ignore	Ignore	Receive	N/A	Drop Packet
Ping	Ignore	Ignore	Ignore	Ignore	Ignore
Invalid	Ignore	Ignore	Ignore	Ignore	Ignore

1. BS Error = Force Bit Stuff Error

   NULL Packet = Zero Length Packet

44.6.6.4 Managing Queue Heads

Figure 44-167 shows the End Point Queue Head.

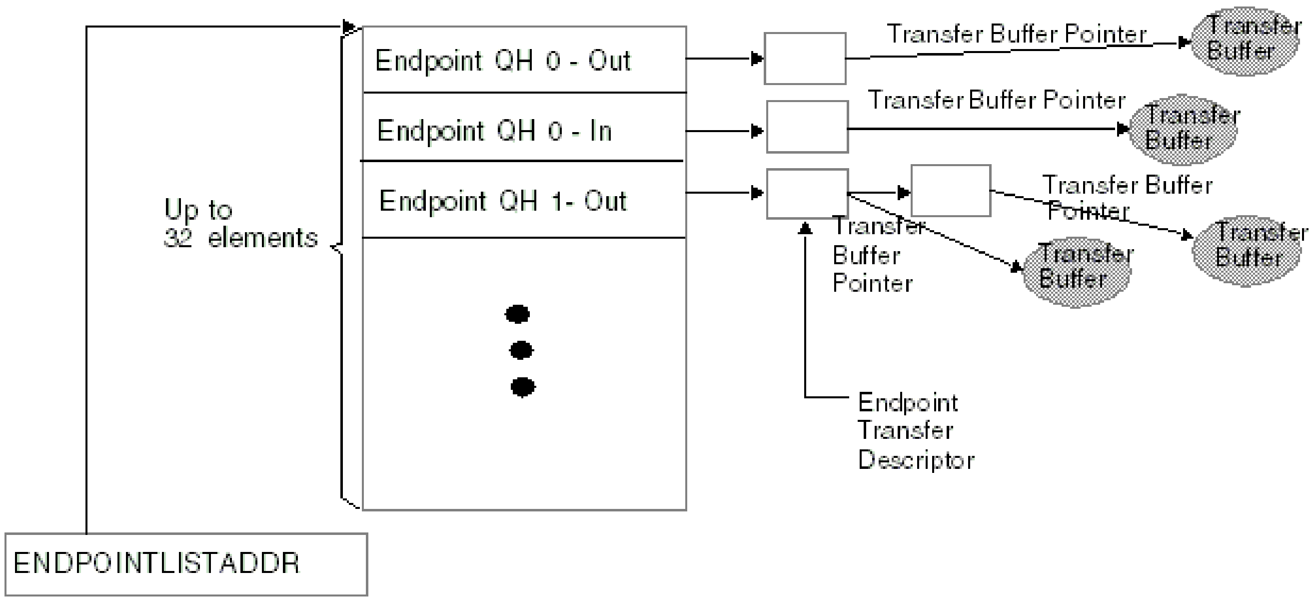


Figure 44-167. End Point Queue Head Diagram

The device queue head (dQH) points to the linked list of transfer tasks, each depicted by the device Transfer Descriptor (dTD). An area of memory pointed to by USB.ENDPOINTLISTADDR contains a group of all dQH's in a sequential list as shown in Figure 44-167. The even elements in the list of dQH's are used for receive endpoints (OUT/SETUP) and the odd elements are used for transmit endpoints (IN/INTERRUPT). Device transfer descriptors are linked head to tail starting at the queue head and ending at a terminate bit. Once the dTD has been retired, it will no longer be part of the linked list from the queue head. Therefore, software is required to track all transfer descriptors because pointers will no longer exist within the queue head once the dTD is retired (see section Software Link Pointers).

In addition to the current and next pointers and the dTD overlay examined in section Operational Model For Packet Transfers, the dQH also contains the following parameters for the associated endpoint: Multipler, Maximum Packet Length, Interrupt On Setup. The complete initialization of the dQH including these fields is demonstrated in the next section.

44.6.6.4.1 Queue Head Initialization

One pair of device queue heads must be initialized for each active endpoint. To initialize a device queue head:

- Write the wMaxPacketSize field as required by the USB Chapter 9 or application specific protocol.
- Write the multiplier field to 0 for control, bulk, and interrupt endpoints. For ISO endpoints, set the multiplier to 1, 2, or 3 as required bandwidth an in conjuction with the USB Chapter 9 protocol.

NOTE

In FS mode, the multiplier field can only be 1 for ISO endpoints.

- Write the next dTD Terminate field to 1.
- Write the Active bit in the status field to 0.
- Write the Halt bit in the status field to 0.

NOTE

The DCD must only modify dQH if the associated endpoint is not primed and there are no outstanding dTD's.

44.6.6.4.2 Operational Model For Setup Transfers

As discussed in section Control Endpoint Operation Model, setup transfer requires special treatment by the DCD. A setup transfer does not use a dTD but instead stores the incoming data from a setup packet in an 8-byte buffer within the dQH.

Upon receiving notification of the setup packet, the DCD should handle the setup transfer as demonstrated here:

1. Copy setup buffer contents from dQH - RX to software buffer.
2. Acknowledge setup backup by writing a "1" to the corresponding bit in ENDPTSETUPSTAT.

NOTE

- The acknowledge must occur before continuing to process the setup packet.
- After the acknowledge has occurred, the DCD must not attempt to access the setup buffer in the dQH - RX. Only the local software copy should be examined.

3. Check for pending data or status dTD's from previous control transfers and flush if any exist as discussed in section Flushing/De-priming an Endpoint.

4. Decode setup packet and prepare data phase [optional] and status phase transfer as required by the USB Chapter 9 or application specific protocol.

NOTE

It is possible for the device controller to receive setup packets before previous control transfers complete. Existing control packets in progress must be flushed and the new control packet completed.

44.6.6.5 Managing Transfers with Transfer Descriptors

44.6.6.5.1 Software Link Pointers

It is necessary for the DCD software to maintain head and tail pointers to the for the linked list of dTDs for each respective queue head. This is necessary because the dQH only maintains pointers to the current working dTD and the next dTD to be executed. The operations described in next section for managing dTD will assume the DCD can use reference the head and tail of the dTD linked list. Figure 44-168 shows the Software Link Pointers.

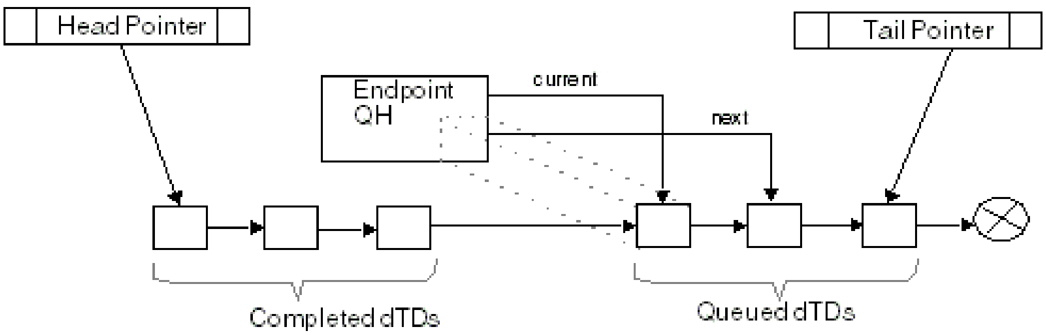


Figure 44-168. Software Link Pointers

NOTE

To conserve memory, the reserved fields at the end of the dQH can be used to store the Head & Tail pointers, but it still remains the responsibility of the DCD to maintain the pointers.

44.6.6.5.2 Building a Transfer Descriptor

Before a transfer can be executed from the linked list, a dTD must be built to describe the transfer. Use the following procedure for building dTDs.

Allocate 8-DWord dTD block of memory aligned to 8-DWord boundaries. Example: bit address 4:0 would be equal to "00000"

Write the following fields:

1. Initialize first 7 DWords to 0.
2. Set the terminate bit to 1.
3. Fill in total bytes with transfer size.
4. Set the interrupt on complete if desired.
5. Initialize the status field with the active bit set to 1 and all remaining status bits set to 0.
6. Fill in buffer pointer page 0 and the current offset to point to the start of the data buffer.
7. Initialize buffer pointer page 1 through page 4 to be one greater than each of the previous buffer pointer.

44.6.6.5.3 Executing A Transfer Descriptor

To safely add a dTD, the DCD must be follow this procedure which will handle the event where the device controller reaches the end of the dTD list at the same time a new dTD is being added to the end of the list.

Determine whether the link list is empty:

- Check DCD driver to see if pipe is empty (internal representation of linked-list should indicate if any packets are outstanding)
- Case 1: Link list is empty
  - 1. Write dQH next pointer AND dQH terminate bit to 0 as a single DWord operation.
  - 2. Clear active & halt bit in dQH (in case set from a previous error).
  - 3. Prime endpoint by writing 1 to correct bit position in Endpoint Initialization (USB_ENDPTPRIME).
- Case 2: Link list is not empty
  - 1. Add dTD to end of linked list.
  - 2. Read correct prime bit in Endpoint Initialization (USB_ENDPTPRIME)- if 1 DONE.
  - 3. Set ATDTW bit in USBCMD register to 1.
  - 4. Read correct status bit in Endpoint Status (USB_ENDPTSTAT). (store in tmp. variable for later)
  - 5. Read ATDTW bit in USBCMD register.
  - If 0 goto 3.
  - If 1 continue to 6.
  - 6. Write ATDTW bit in USBCMD register to 0.

- 7. If status bit read in (3) is 1 DONE.
- 8. If status bit read in (3) is 0 then Goto Case 1: Step 1.

44.6.6.5.4 Transfer Completion

After a dTD has been initialized and the associated endpoint primed the device controller will execute the transfer upon the host-initiated request. The DCD will be notified with a USB interrupt if the Interrupt On Complete bit was set or alternately, the DCD can poll the endpoint complete register to find when the dTD had been executed. After a dTD has been executed, DCD can check the status bits to determine success or failure.

NOTE

Multiple dTD can be completed in a single endpoint complete notification. After clearing the notification, DCD must search the dTD linked list and retire all dTDs that have finished (Active bit cleared).

By reading the status fields of the completed dTDs, the DCD can determine if the transfers completed successfully. Success is determined with the following combination of status bits:

- Active = 0
- Halted = 0
- Transaction Error = 0
- Data Buffer Error = 0

Should any combination other than the one shown above exist, the DCD must take proper action. Transfer failure mechanisms are indicated in the Device Error Matrix.

In addition to checking the status bit the DCD must read the Transfer Bytes field to determine the actual bytes transferred. When a transfer is complete, the Total Bytes transferred is by decremented by the actual bytes transferred. For Transmit packets, a packet is only complete after the actual bytes reaches zero, but for receive packets, the host may send fewer bytes in the transfer according the USB variable length packet protocol.

44.6.6.5.5 Flushing/De-priming an Endpoint

It is necessary for the DCD to flush to de-prime one more endpoints on a USB device reset or during a broken control transfer. There may also be application specific requirements to stop transfers in progress. The following procedure can be used by the DCD to stop a transfer in progress:

1. Write a '1' to the corresponding bit(s) in Endpoint De-Initialize (USB_ENDPTFLUSH).
2. Wait until all bits in Endpoint De-Initialize (USB_ENDPTFLUSH) are '0'.
   - Software note: this operation may take a large amount of time depending on the USB bus activity. It is not desirable to have this wait loop within an interrupt service routine.
3. Read Endpoint Status (USB_ENDPTSTAT) to ensure that for all endpoints commanded to be flushed, that the corresponding bits are now '0'. If the corresponding bits are '1' after step #2 has finished, then the flush failed as described in the following:
   - Explanation: In very rare cases, a packet is in progress to the particular endpoint when commanded flush using Endpoint De-Initialize (USB_ENDPTFLUSH). A safeguard is in place to refuse the flush to ensure that the packet in progress completes successfully. The DCD may need to repeatedly flush any endpoints that fail to flush be repeating steps 1-3 until each endpoint is successfully flushed.

44.6.6.5.6 Device Error Matrix

Table 44-215 summarizes packet errors that are not automatically handled by the Device Controller.

Table 44-215. Device Error Matrix

Error	Direction	Packet Type	Data Buffer Error Bit	Transaction Error Bit
Overflow **	RX	Any	1	0
ISO Packet Error	RX	ISO	0	1
ISO Fulfillment Error	Both	ISO	0	1

Notice that the device controller handles all errors on Bulk/Control/Interrupt Endpoints except for a data buffer overflow. However, for ISO endpoints, errors packets are not retried and errors are tagged as indicated. Table 44-216 describes the errors.

Table 44-216. Error Descriptions

Error	Description
Overflow	Number of bytes received exceeded max. packet size or total buffer length.
	** This error will also set the Halt bit in the dQH and if there are dTDs remaining in the linked list for the endpoint, then those will not be executed.
ISO Packet Error	CRC Error on received ISO packet. Contents not guaranteed to be correct.
ISO Fulfillment Error	Hst failed to complete the number of packets defined in the dQH mult field within the given (micro)frame. For scheduled data delivery the DCD may need to readjust the data queue because a fulfillment error will cause Device Controller to cease data transfers on the pipe for one (micro)frame. During the "dead" (micro)frame, the Device Controller reports error on the pipe and primes for the following frame.

44.6.6.6 Servicing Interrupts

The interrupt service routine must consider that there are high-frequency, low-frequency operations, and error operations and order accordingly.

44.6.6.6.1 High-Frequency Interrupts

High frequency interrupts in particular should be handed in the order below. The most important of these is listed first because the DCD must acknowledge a setup buffer in the timeliest manner possible.

Table 44-217 describes the High frequency interrupt events.

Table 44-217. High Frequency Interrupt Events

Execution Order	Interrupt	Action
1a	USB Interrupt - USB.ENDPTSETUPSTATUS	Copy contents of setup buffer and acknowledge setup packet (as indicated in Figure 44-167 shows the End Point Queue Head). Process setup packet according to USB 2.0 Chapter 9 or application specific protocol.
1b	USB Interrupt[1] - USB.ENDPTCOMPLETE	Handle completion of dTD as indicated in Figure 44-167 shows the End Point Queue Head.
2	SOF Interrupt	Action as deemed necessary by application. This interrupt may not have a use in all applications.

1. It is likely that multiple interrupts to stack up on any call to the Interrupt Service Routine AND during the Interrupt Service Routine.

44.6.6.6.2 Low-Frequency Interrupts

The low frequency events include the following interrupts. These interrupt can be handled in any order because they do not occur often in comparison to the high-frequency interrupts.

Table 44-218 shows the Low frequency interrupt events.

Table 44-218. Low Frequency Interrupt Events

Interrupt	Action
Port Change	Change software state information.
Sleep Enable (Suspend)	Change software state information. Low power handling as necessary.
Reset Received	Change software state information. Abort pending transfers.

44.6.6.6.3 Error Interrupts

Error interrupts will be least frequent and should be placed last in the interrupt service routine.

Table 44-219 shows the error interrput events

.

Table 44-219. Error Interrupt Events

Interrupt	Action
USB Error Interrupt	This error is redundant because it combines USB Interrupt and an error status in the dTD. The DCD will more aptly handle packet-level errors by checking dTD status field upon receipt of USB Interrupt (w/ USB.ENDPTCOMPLETE).
System Error	Unrecoverable error. Immediate Reset of core; free transfers buffers in progress and restart the DCD.

44.7 Glossary of Terms and Abbreviations

This chapter lists and defines terms and abbreviations used throughout this specification.

Table 44-220. Glossary

ACK	Handshake packet indicating a positive acknowledgment.
Active Device	A device that is powered and is not in the Suspend state.
Asynchronous Data	Data transferred at irregular intervals with relaxed latency requirements.
Asynchronous RA	The incoming data rate, $F^{s}i$, and the outgoing data rate, $F^{s}o$, of the RA process are independent (i.e., there is no shared master clock). See also rate adaptation.
Asynchronous SRC	The incoming sample rate, $F^{s}i$, and outgoing sample rate, $F^{s}o$, of the SRC process are independent (i.e., there is no shared master clock). See also sample rate conversion.
Audio Device	A device that sources or sinks sampled analog data.
AWG#	The measurement of a wire's cross-section, as defined by the American Wire Gauge standard.
b/s	Transmission rate expressed in bits per second.
B/s	Transmission rate expressed in bytes per second.
Babble	Unexpected bus activity that persists beyond a specified point in a (micro) frame.
Bandwidth	The amount of data transmitted per unit of time, typically bits per second (b/s) or bytes per second (B/s).
Big Endian	A method of storing data that places the most significant byte of multiple-byte values at a lower storage address. For example, a 16-bit integer stored in big endian format places the least significant byte at the higher address and the most significant byte at the lower address. See also little endian.
Bit	A unit of information used by digital computers. Represents the smallest piece of addressable memory within a computer. A bit expresses the choice between two possibilities and is typically represented by a logical one (1) or zero (0).

Table continues on the next page...

Table 44-220. Glossary (continued)

Bit Stuffing	Insertion of a "0" bit into a data stream to cause an electrical transition on the data wires, allowing a PLL to remain locked.
Buffer	Storage used to compensate for a difference in data rates or time of occurrence of events, when transmitting data from one device to another.
Bulk Transfer	One of the four USB transfer types. Bulk transfers are non-periodic, large burst communication typically used for a transfer that can use any available bandwidth and can be delayed until bandwidth is available. See also transfer type.
Bus Enumeration	Detecting and identifying USB devices.
Byte	A data element that is eight bits in size.
Capabilities	Those attributes of a USB device that are administrated by the host.
Characteristics	Those qualities of a USB device that are unchangeable; for example, the device class is a device characteristic.
Client	Software resident on the host that interacts with the USB System Software to arrange data transfer between a function and the host. The client is often the data provider and consumer for transferred data.
Configuring Software	Software resident on the host software that is responsible for configuring a USB device. This may be a system configurator or software specific to the device.
Control Endpoint	A pair of device endpoints with the same endpoint number that are used by a control pipe. Control endpoints transfer data in both directions and, therefore, use both endpoint directions of a device address and endpoint number combination. Thus, each control endpoint consumes two endpoint addresses.
Control Pipe	Same as a message pipe.
Control Transfer	One of the four USB transfer types. Control transfers support configuration/command/status type communications between client and function. See also transfer type.
CRC	See Cyclic Redundancy Check.
CTI	Computer Telephony Integration.
Cyclic Redundancy Check (CRC)	A check performed on data to see if an error has occurred in transmitting, reading, or writing the data. The result of a CRC is typically stored or transmitted with the checked data. The stored or transmitted result is compared to a CRC calculated for the data to determine if an error has occurred.
Default Address	An address defined by the USB Specification and used by a USB device when it is first powered or reset. The default address is 00H.
Default Pipe	The message pipe created by the USB System Software to pass control and status information between the host and a USB device's endpoint zero.
Device	A logical or physical entity that performs a function. The actual entity described depends on the context of the reference. At the lowest level, device may refer to a single hardware component, as in a memory device. At a higher level, it may refer to a collection of hardware components that perform a particular function, such as a USB interface device. At an even higher level, device may refer to the function performed by an entity attached to the USB; for example, a data/FAX modem device. Devices may be physical, electrical, addressable, and logical. When used as a non-specific reference, a USB device is either a hub or a function.
Device Address	A seven-bit value representing the address of a device on the USB. The device address is the default address (00H) when the USB device is first powered or the device is reset. Devices are assigned a unique device address by the USB System Software.
Device Endpoint	A uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between the host and device. See also endpoint address.
Device Resources	Resources provided by USB devices, such as buffer space and endpoints. See also Host Resources and Universal Serial Bus Resources.

Table continues on the next page...

Table 44-220. Glossary (continued)

Device Software	Software that is responsible for using a USB device. This software may or may not also be responsible for configuring the device for use.
Downstream	The direction of data flow from the host or away from the host. A downstream port is the port on a hub electrically farthest from the host that generates downstream data traffic from the hub. Downstream ports receive upstream data traffic.
Driver	When referring to hardware, an I/O pad that drives an external load. When referring to software, a program responsible for interfacing to a hardware device, that is, a device driver.
DWord	Double word. A data element that is two words (i.e., four bytes or 32 bits) in size.
Dynamic Insertion and Removal	The ability to attach and remove devices while the host is in operation.
E^2PROM	See Electrically Erasable Programmable Read Only Memory.
EEPROM	See Electrically Erasable Programmable Read Only Memory.
Electrically Erasable Programmable Read Only Memory (EEPROM)	Non-volatile rewriteable memory storage technology.
End User	The user of a host.
Endpoint	See device endpoint.
Endpoint Address	The combination of an endpoint number and an endpoint direction on a USB device. Each endpoint address supports data transfer in one direction.
Endpoint Direction	The direction of data transfer on the USB. The direction can be either IN or OUT. IN refers to transfers to the host; OUT refers to transfers from the host.
Endpoint Number	A four-bit value between 0H and FH, inclusive, associated with an endpoint on a USB device.
Envelope detector	An electronic circuit inside a USB device that monitors the USB data lines and detects certain voltage related signal characteristics.
EOF	End-of- (micro) Frame.
EOP	End-of-Packet.
External Port	See port.
Eye pattern	A representation of USB signaling that provides minimum and maximum voltage levels as well as signal jitter.
False EOP	A spurious, usually noise-induced event that is interpreted by a packet receiver as an EOP.
Flush (Endpoint)	A term used in this device controller implementation to describe the action of clearing an endpoint ready status.
Frame	A 1-millisecond time base established on full-/low-speed buses.
Frame Pattern	A sequence of frames that exhibit a repeating pattern in the number of samples transmitted per frame. For a 44.1 kHz audio transfer, the frame pattern could be nine frames containing 44 samples followed by one-frame containing45 samples.
Fs	See sample rate.
Full-duplex	Computer data transmission occurring in both directions simultaneously.
Full-speed	USB operation at 12 Mb/s. See also low-speed and high-speed
Function	A USB device that provides a capability to the host, such as an ISDN connection, a digital microphone, or speakers.
Handshake Packet	A packet that acknowledges or rejects a specific condition. For examples, see ACK and NAK.
High-bandwidth endpoint	A high-speed device endpoint that transfers more than 1024 bytes and less than3073 bytes per microframe.
High-speed	USB operation at 480 Mb/s. See also low-speed and full-speed

Table continues on the next page...

Table 44-220. Glossary (continued)

Host	The host computer system where the USB Host Controller is installed. This includes the host hardware platform (CPU, bus, etc.) and the operating system in use.
Host Controller	The host's USB interface.
Host Controller Driver (HCD)	The USB software layer that abstracts the Host Controller hardware. The Host Controller Driver provides an SPI for interaction with a Host Controller. The Host Controller Driver hides the specifics of the Host Controller hardware implementation.
Host Resources	Resources provided by the host, such as buffer space and interrupts. See also Device Resources and Universal Serial Bus Resources.
Hub	A USB device that provides additional connections to the USB.
Hub Tier	One plus the number of USB links in a communication path between the host and a function.
I/O Request Packet	An identifiable request by a software client to move data between itself (on the host) and an endpoint of a device in an appropriate direction.
Interrupt Request (IRQ)	A hardware signal that allows a device to request attention from a host. The host typically invokes an interrupt service routine to handle the condition that caused the request.
Interrupt Transfer	One of the four USB transfer types. Interrupt transfer characteristics are small data, non-periodic, low frequency, and bounded-latency. Interrupt transfers are typically used to handle service needs. See also transfer type.
IRP	See I/O Request Packet.
IRQ	See Interrupt Request.
Isochronous Data	A stream of data whose timing is implied by its delivery rate
Isochronous Device	An entity with isochronous endpoints, as defined in the USB Specification, that sources or sinks sampled analog streams or synchronous data streams.
Isochronous Sink Endpoint	An endpoint that is capable of consuming an isochronous data stream that is sent by the host.
Isochronous Source Endpoint	An endpoint that is capable of producing an isochronous data stream and sending it to the host.
Isochronous Transfer	One of the four USB transfer types. Isochronous transfers are used when working with isochronous data. Isochronous transfers provide periodic, continuous communication between host and device. See also transfer type.
Jitter	A tendency toward lack of synchronization caused by mechanical or electrical changes. More specifically, the phase shift of digital pulses over a transmission medium.
kb/s	Transmission rate expressed in kilobits per second.
kB/s	Transmission rate expressed in kilobytes per second.
Little Endian	Method of storing data that places the least significant byte of multiple-byte values at lower storage addresses. For example, a 16-bit integer stored in little endian format places the least significant byte at the lower address and the most significant byte at the next address. See also big endian.
LOA	Loss of bus activity characterized by an SOP without a corresponding EOP.
Low-speed	USB operation at 1.5 Mb/s. See also full-speed and high-speed.
LSb	Least significant bit.
LSB	Least significant byte.
Mb/s	Transmission rate expressed in megabits per second.
MB/s	Transmission rate expressed in megabytes per second.
Message Pipe	A bi-directional pipe that transfers data using a request/data/status paradigm. The data has an imposed structure that allows requests to be reliably identified and communicated.
Microframe	A 125-microsecond time base established on high-speed buses.

Table continues on the next page...

Table 44-220. Glossary (continued)

MSb	Most significant bit.
MSB	Most significant byte.
NAK	Handshake packet indicating a negative acknowledgment.
Non Return to Zero Invert (NRZI)	A method of encoding serial data in which ones and zeroes are represented by opposite and alternating high and low voltages where there is no return to zero (reference) voltage between encoded bits. Eliminates the need for clock pulses.
NRZI	See Non Return to Zero Invert.
Object	Host software or data structure representing a USB entity.
Packet	A bundle of data organized in a group for transmission. Packets typically contain three elements: control information (e.g., source, destination, and length), the data to be transferred, and error detection and correction bits.
Packet Buffer	The logical buffer used by a USB device for sending or receiving a single packet. This determines the maximum packet size the device can send or receive.
Packet ID (PID)	A field in a USB packet that indicates the type of packet and by inference, the format of the packet and the type of error detection applied to the packet.
PCI	Programming and control interface
Phase	A token, data, or handshake packet. A transaction has three phases
Phase Locked Loop (PLL)	A circuit that acts as a phase detector to keep an oscillator in phase with an incoming frequency.
Physical Device	A device that has a physical implementation; e.g., speakers, microphones, and CD players.
PID	See Packet ID.
Pipe	A logical abstraction representing the association between an endpoint on a device and software on the host. A pipe has several attributes; for example, a pipe may transfer data as streams (stream pipe) or messages (message pipe) See also stream pipe and message pipe.
PLL	See Phase Locked Loop.
Polling	Asking multiple devices, one at a time, if they have any data to transmit.
POR	See Power On Reset.
Port	Point of access to or from a system or circuit. For the USB, the point where a USB device is attached.
Power On Reset (POR)	Restoring a storage device, register, or memory to a predetermined state when power is applied.
Prime (Endpoint)	A term used in this device controller implementation to describe the action of readying an endpoint to transmit or receive data.
Programmable Data Rate	A fixed data rate (single-frequency endpoints), a limited number of data rates (32 kHz, 44.1 kHz, 48 kHz, …), or a continuously programmable data rate. The exact programming capabilities of an endpoint must be reported in the appropriate class-specific endpoint descriptors.
Protocol	A specific set of rules, procedures, or conventions relating to format and timing of data transmission between two devices.
RA	See rate adaptation.
Rate Adaptation	The process by which an incoming data stream, sampled at $F^{s}i$, is converted to an outgoing data stream, sampled at $F^{s}o$, with a certain loss of quality, determined by the rate adaptation algorithm. Error control mechanisms are required for the process. $F^{s}i$ and $F^{s}o$ can be different and asynchronous. $F^{s}i$ is the input data rate of the RA; $F^{s}o$ is the output data rate of the RA.
Request	A request made to a USB device contained within the data portion of a SETUP packet.
Retire	The action of completing service for a transfer and notifying the appropriate software client of the completion.
Root Hub	A USB hub directly attached to the Host Controller. This hub (tier 1) is attached to the host.

Table continues on the next page...

Table 44-220. Glossary (continued)

Root Port	The downstream port on a Root Hub.
Sample	The smallest unit of data on which an endpoint operates; a property of an endpoint.
Sample Rate (Fs)	The number of samples per second, expressed in Hertz (Hz).
Sample Rate Conversion (SRC)	A dedicated implementation of the RA process for use on sampled analog data streams. The error control mechanism is replaced by interpolating techniques.
Service	A procedure provided by a System Programming Interface (SPI).
Service Interval	The period between consecutive requests to a USB endpoint to send or receive data.
Service Jitter	The deviation of service delivery from its scheduled delivery time.
Service Rate	The number of services to a given endpoint per unit time.
SOF	See Start-of-Frame.
SOP	Start-of-Packet.
SPI	See System Programming Interface.
Split transaction	A transaction type supported by host controllers and hubs. This transaction type allows full- and low-speed devices to be attached to hubs operating at high-speed.
SRC	See Sample Rate Conversion.
Stage	One part of the sequence composing a control transfer; stages include the Setup stage, the Data stage, and the Status stage.
Start-of-Frame (SOF)	The first transaction in each (micro) frame. An SOF allows endpoints to identify the start of the (micro) frame and synchronize internal endpoint clocks to the host.
Stream Pipe	A pipe that transfers data as a stream of samples with no defined USB structure.
Synchronization Type	A classification that characterizes an isochronous endpoint's capability to connect to other isochronous endpoints.
Synchronous RA	The incoming data rate, F^si, and the outgoing data rate, F^so, of the RA process, are derived from the same master clock. There is a fixed relation between F^si and F^so.
Synchronous SRC	The incoming sample rate, F^si, and outgoing sample rate, F^so, of the SRC process are derived from the same master clock. There is a fixed relation between F^si and F^so.
System Programming Interface (SPI)	A defined interface to services provided by system software.
TDM	See Time Division Multiplexing.
TDR	See Time Domain Reflectometer.
Termination	Passive components attached at the end of cables to prevent signals from being reflected or echoed.
Time Division Multiplexing (TDM)	A method of transmitting multiple signals (data, voice, and/or video) simultaneously over one communications medium by interleaving a piece of each signal one after another.
Time Domain Reflectometer (TDR)	An instrument capable of measuring impedance characteristics of the USB signal lines.
Timeout	The detection of a lack of bus activity for some predetermined interval.
Token Packet	A type of packet that identifies what transaction is to be performed on the bus.
Transaction	The delivery of service to an endpoint; consists of a token packet, optional data packet, and optional handshake packet. Specific packets are allowed/required based on the transaction type.
Companion Controller	A functional component of a USB hub. The Companion Controller responds to special high-speed transactions and translates them to full/low-speed transactions with full/low-speed devices attached on downstream facing ports.
Transfer	One or more bus transactions to move information between a software client and its function.

Table continues on the next page...

Table 44-220. Glossary (continued)

Transfer Type	Determines the characteristics of the data flow between a software client and its function. Four standard transfer types are defined: control, interrupt, bulk, and isochronous.
Turn-around Time	The time a device needs to wait to begin transmitting a packet after a packet has been received to prevent collisions on the USB. This time is based on the length and propagation delay characteristics of the cable and the location of the transmitting device in relation to other devices on the USB.
Universal Serial Bus Driver (USBD)	The host resident software entity responsible for providing common services to clients that are manipulating one or more functions on one or more Host Controllers.
Universal Serial Bus Resources	Resources provided by the USB, such as bandwidth and power. See also Device Resources and Host Resources.
Upstream	The direction of data flow towards the host. An upstream port is the port on a device electrically closest to the host that generates upstream data traffic from the hub. Upstream ports receive downstream data traffic.
USBD	See Universal Serial Bus Driver.
USB-IF	USB Implementers Forum, Inc. is a nonprofit corporation formed to facilitate the development of USB compliant products and promote the technology.
Virtual Device	A device that is represented by a software interface layer. An example of a virtual device is a hard disk with its associated device driver and client software that makes it able to reproduce an audio .WAV file.
Word	A data element that is two bytes (16 bits) in size.

Chapter 45
Universal Serial Bus 2.0 Integrated PHY (USBPHY)

45.1 USB PHY Overview

The chip contains 2 integrated USB 2.0 PHY macrocells capable of connecting to USB host/device systems at the USB low-speed (LS) rate of 1.5 Mbits/s, full-speed (FS) rate of 12 Mbits/s or at the USB 2.0 high-speed (HS) rate of 480 Mbits/s. See the following figure for a block diagram of the PHY.

The integrated PHY provides a standard UTM interface. The USB_DP and USB_DM pins connect directly to a USB connector. USBPHY0 is the PHY interface for USB OTG 0 controller; USBPHY1 is the PHY interface for USB OTG 1 controller.

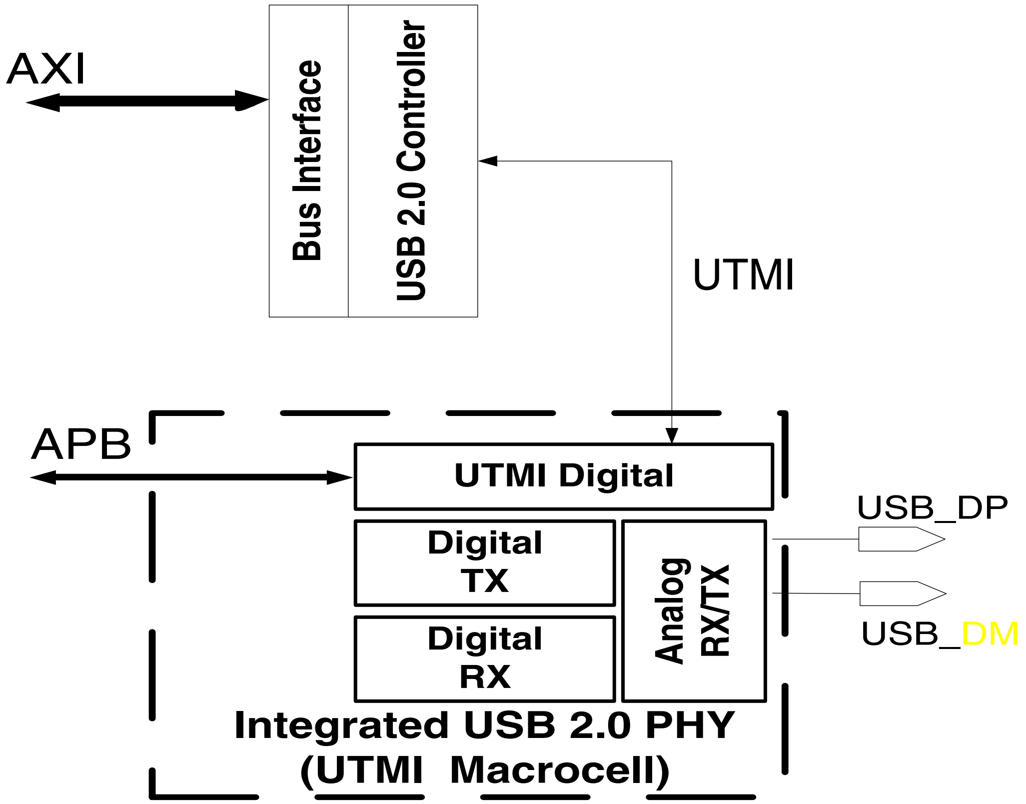


Figure 45-1. USB 2.0 PHY Block Diagram

The following subsections describe the external interfaces, internal interfaces, major blocks, and programmable registers that comprise the integrated USB 2.0 PHY.

45.2 USB PHY Memory Map/Register Definition

USBPHY Hardware Register Format Summary

USBPHY0 base address is 0x 4005_ 0800; USBPHY1 base address is 0x 4005_ 0C00

USBPHY memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_0800	USB PHY Power-Down Register (USBPHY0_PWD)	32	R/W	001E_1C00h	45.2.1/2375

Table continues on the next page...

USBPHY memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_0804	USB PHY Power-Down Register (USBPHY0_PWD_SET)	32	R/W	001E_1C00h	45.2.1/2375
4005_0808	USB PHY Power-Down Register (USBPHY0_PWD_CLR)	32	R/W	001E_1C00h	45.2.1/2375
4005_080C	USB PHY Power-Down Register (USBPHY0_PWD_TOG)	32	R/W	001E_1C00h	45.2.1/2375
4005_0810	USB PHY Transmitter Control Register (USBPHY0_TX)	32	R/W	1006_0607h	45.2.2/2377
4005_0820	USB PHY Receiver Control Register (USBPHY0_RX)	32	R/W	0000_0000h	45.2.3/2378
4005_0824	USB PHY Receiver Control Register (USBPHY0_RX_SET)	32	R/W	0000_0000h	45.2.3/2378
4005_0828	USB PHY Receiver Control Register (USBPHY0_RX_CLR)	32	R/W	0000_0000h	45.2.3/2378
4005_082C	USB PHY Receiver Control Register (USBPHY0_RX_TOG)	32	R/W	0000_0000h	45.2.3/2378
4005_0830	USB PHY General Control Register (USBPHY0_CTRL)	32	R/W	C020_0040h	45.2.4/2380
4005_0834	USB PHY General Control Register (USBPHY0_CTRL_SET)	32	R/W	C020_0040h	45.2.4/2380
4005_0838	USB PHY General Control Register (USBPHY0_CTRL_CLR)	32	R/W	C020_0040h	45.2.4/2380
4005_083C	USB PHY General Control Register (USBPHY0_CTRL_TOG)	32	R/W	C020_0040h	45.2.4/2380
4005_0840	USB PHY Status Register (USBPHY0_STATUS)	32	R/W	0000_0000h	45.2.5/2383
4005_0850	USB PHY Debug Register (USBPHY0_DEBUG)	32	R/W	7F18_0000h	45.2.6/2385
4005_0854	USB PHY Debug Register (USBPHY0_DEBUG_SET)	32	R/W	7F18_0000h	45.2.6/2385
4005_0858	USB PHY Debug Register (USBPHY0_DEBUG_CLR)	32	R/W	7F18_0000h	45.2.6/2385
4005_085C	USB PHY Debug Register (USBPHY0_DEBUG_TOG)	32	R/W	7F18_0000h	45.2.6/2385
4005_0860	UTMI Debug Status Register 0 (USBPHY0_DEBUG0_STATUS)	32	R	0000_0000h	45.2.7/2387
4005_0870	UTMI Debug Status Register 1 (USBPHY0_DEBUG1)	32	R/W	0000_1000h	45.2.8/2387
4005_0874	UTMI Debug Status Register 1 (USBPHY0_DEBUG1_SET)	32	R/W	0000_1000h	45.2.8/2387
4005_0878	UTMI Debug Status Register 1 (USBPHY0_DEBUG1_CLR)	32	R/W	0000_1000h	45.2.8/2387
4005_087C	UTMI Debug Status Register 1 (USBPHY0_DEBUG1_TOG)	32	R/W	0000_1000h	45.2.8/2387
4005_0880	UTMI RTL Version (USBPHY0_VERSION)	32	R	0402_0000h	45.2.9/2388
4005_0890	USB PHY IP Block Register (USBPHY0_IP)	32	R/W	0000_0000h	45.2.10/ 2389
4005_0894	USB PHY IP Block Register (USBPHY0_IP_SET)	32	R/W	0000_0000h	45.2.10/ 2389
4005_0898	USB PHY IP Block Register (USBPHY0_IP_CLR)	32	R/W	0000_0000h	45.2.10/ 2389
4005_089C	USB PHY IP Block Register (USBPHY0_IP_TOG)	32	R/W	0000_0000h	45.2.10/ 2389
4005_0C00	USB PHY Power-Down Register (USBPHY1_PWD)	32	R/W	001E_1C00h	45.2.1/2375
4005_0C04	USB PHY Power-Down Register (USBPHY1_PWD_SET)	32	R/W	001E_1C00h	45.2.1/2375
4005_0C08	USB PHY Power-Down Register (USBPHY1_PWD_CLR)	32	R/W	001E_1C00h	45.2.1/2375
4005_0C0C	USB PHY Power-Down Register (USBPHY1_PWD_TOG)	32	R/W	001E_1C00h	45.2.1/2375
4005_0C10	USB PHY Transmitter Control Register (USBPHY1_TX)	32	R/W	1006_0607h	45.2.2/2377

Table continues on the next page...

USBPHY memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_0C20	USB PHY Receiver Control Register (USBPHY1_RX)	32	R/W	0000_0000h	45.2.3/2378
4005_0C24	USB PHY Receiver Control Register (USBPHY1_RX_SET)	32	R/W	0000_0000h	45.2.3/2378
4005_0C28	USB PHY Receiver Control Register (USBPHY1_RX_CLR)	32	R/W	0000_0000h	45.2.3/2378
4005_0C2C	USB PHY Receiver Control Register (USBPHY1_RX_TOG)	32	R/W	0000_0000h	45.2.3/2378
4005_0C30	USB PHY General Control Register (USBPHY1_CTRL)	32	R/W	C020_0040h	45.2.4/2380
4005_0C34	USB PHY General Control Register (USBPHY1_CTRL_SET)	32	R/W	C020_0040h	45.2.4/2380
4005_0C38	USB PHY General Control Register (USBPHY1_CTRL_CLR)	32	R/W	C020_0040h	45.2.4/2380
4005_0C3C	USB PHY General Control Register (USBPHY1_CTRL_TOG)	32	R/W	C020_0040h	45.2.4/2380
4005_0C40	USB PHY Status Register (USBPHY1_STATUS)	32	R/W	0000_0000h	45.2.5/2383
4005_0C50	USB PHY Debug Register (USBPHY1_DEBUG)	32	R/W	7F18_0000h	45.2.6/2385
4005_0C54	USB PHY Debug Register (USBPHY1_DEBUG_SET)	32	R/W	7F18_0000h	45.2.6/2385
4005_0C58	USB PHY Debug Register (USBPHY1_DEBUG_CLR)	32	R/W	7F18_0000h	45.2.6/2385
4005_0C5C	USB PHY Debug Register (USBPHY1_DEBUG_TOG)	32	R/W	7F18_0000h	45.2.6/2385
4005_0C60	UTMI Debug Status Register 0 (USBPHY1_DEBUG0_STATUS)	32	R	0000_0000h	45.2.7/2387
4005_0C70	UTMI Debug Status Register 1 (USBPHY1_DEBUG1)	32	R/W	0000_1000h	45.2.8/2387
4005_0C74	UTMI Debug Status Register 1 (USBPHY1_DEBUG1_SET)	32	R/W	0000_1000h	45.2.8/2387
4005_0C78	UTMI Debug Status Register 1 (USBPHY1_DEBUG1_CLR)	32	R/W	0000_1000h	45.2.8/2387
4005_0C7C	UTMI Debug Status Register 1 (USBPHY1_DEBUG1_TOG)	32	R/W	0000_1000h	45.2.8/2387
4005_0C80	UTMI RTL Version (USBPHY1_VERSION)	32	R	0402_0000h	45.2.9/2388
4005_0C90	USB PHY IP Block Register (USBPHY1_IP)	32	R/W	0000_0000h	45.2.10/ 2389
4005_0C94	USB PHY IP Block Register (USBPHY1_IP_SET)	32	R/W	0000_0000h	45.2.10/ 2389
4005_0C98	USB PHY IP Block Register (USBPHY1_IP_CLR)	32	R/W	0000_0000h	45.2.10/ 2389
4005_0C9C	USB PHY IP Block Register (USBPHY1_IP_TOG)	32	R/W	0000_0000h	45.2.10/ 2389

45.2.1 USB PHY Power-Down Register (USBPHYx_PWD*n*)

The USB PHY Power-Down Register provides overall control of the PHY power state.

Address: Base address + 0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD2											RXPWDRX	RXPWDDIFF	RXPWD1PT1	RXPWDENV	RSVD1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1			TXPWDV2I	TXPWDIBIAS	TXPWDFS	RSVD0									
W																
Reset	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0

USBPHYx_PWD*n* field descriptions

Field	Description
31–21 RSVD2	Reserved.
20 RXPWDRX	0 = Normal operation. 1 = Power-down the entire USB PHY receiver block except for the full-speed differential receiver. Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHY_CTRL is enabled.
19 RXPWDDIFF	0 = Normal operation. 1 = Power-down the USB high-speed differential receiver. Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHY_CTRL is enabled.

Table continues on the next page...

USBPHYx_PWD*n* field descriptions (continued)

Field	Description
18 RXPWD1PT1	0 = Normal operation. 1 = Power-down the USB full-speed differential receiver. Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHY_CTRL is enabled.
17 RXPWDENV	0 = Normal operation. 1 = Power-down the USB high-speed receiver envelope detector (squelch signal). Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHY_CTRL is enabled.
16–13 RSVD1	Reserved.
12 TXPWDV2I	This bit is used to powerdown the USB PHY transmit V-to-I converter and the current mirror. **NOTE:** This bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHY_CTRL is enabled. 0 Nornal operation. 1 Powerdown the USB PHY transmit V-to-I converter and the current mirror.
11 TXPWDIBIAS	This bit can power-down the USB PHY current bias block for the transmitter. **NOTE:** This bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHY_CTRL is enabled. 0 Normal operation 1 Power-down the USB PHY current bias block for the transmitter. This bit should be set only when the USB is in suspend mode. This effectively powers down the entire USB transmit path.
10 TXPWDFS	0 = Normal operation. 1 = Power-down the USB full-speed drivers. This turns off the current starvation sources and puts the drivers into high-impedance output. Note that this bit will be auto cleared if there is USB wakeup event while ENAUTOCLR_PHY_PWD bit of USBPHY_CTRL is enabled.
9–0 RSVD0	Reserved.

45.2.2 USB PHY Transmitter Control Register (USBPHYx_TX)

The USB PHY Transmitter Control Register handles the transmit controls.

Address: Base address + 10h offset = 4005_0810h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD2												TXCAL45DP				RSVD1				TXCAL45DM				RSVD0				D_CAL			
W																																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1

USBPHYx_TX field descriptions

Field	Description
31–20 RSVD2	Reserved.
19–16 TXCAL45DP	Decode to trim the nominal 45-Ohm resistance to the USB_DP output pin. Maximum resistance = 0000. Resistance is centered by design at 0110. **NOTE:** Trimming this resistance will impact both the overshoot/undershoot of the Full Speed TX output and the amplitude of the High Speed TX output.
15–12 RSVD1	Reserved. **Note:** This bit should remain clear.
11–8 TXCAL45DM	Decode to trim the nominala 45-Ohm resistance to the USB_DM output pin. Maximum resistance = 0000. Resistance is centered by design at 0110.
7–4 RSVD0	Reserved. **Note:** This bit should remain clear.
3–0 D_CAL	Decode to trim the nominal 17.78ma current source for the High Speed TX drivers on USB_DP and USB_DM. This current is directly proportional to the amplitude of the High Speed TX eye diagram.> 0000 Maximum current, approximately 19% above nominal. 0111 Nominal. 1111 Minimum current, approximately 19% below nominal

45.2.3 USB PHY Receiver Control Register (USBPHY*x*_RX*n*)

The USB PHY Receiver Control Register handles receive path controls.

Address: Base address + 20h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD2									RXDBYPASS	RSVD1					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1									DISCONADJ			RSVD0	ENVADJ		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBPHY*x*_RX*n* field descriptions

Field	Description
31–23 RSVD2	Reserved.
22 RXDBYPASS	0 = Normal operation. 1 = Use the output of the USB_DP single-ended receiver in place of the full-speed differential receiver. This test mode is intended for lab use only.

Table continues on the next page...

USBPHYx_RX*n* field descriptions (continued)

Field	Description
21–7 RSVD1	Reserved.
6–4 DISCONADJ	The DISCONADJ field adjusts the trip point for the disconnect detector: 0000 = Trip-Level Voltage is 0.57500 V 0001 = Trip-Level Voltage is 0.56875 V 0010 = Trip-Level Voltage is 0.58125 V 0011 = Trip-Level Voltage is 0.58750 V 01XX = Reserved 1XXX = Reserved
3 RSVD0	Reserved.
2–0 ENVADJ	The ENVADJ field adjusts the trip point for the envelope detector. 000 = Trip-Level Voltage is 0.11250 V 001 = Trip-Level Voltage is 0.10000 V 010 = Trip-Level Voltage is 0.13750 V 011 = Trip-Level Voltage is 0.10625 V 1XX = Reserved

45.2.4 USB PHY General Control Register (USBPHYx_CTRLn)

The USB PHY General Control Register handles OTG and Host controls. This register also includes interrupt enables and connectivity detect enables and results.

Address: Base address + 30h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	SFTRST	CLKGATE	UTMI_SUSPENDM	HOST_FORCE_LS_SE0	OTG_ID_VALUE	Reserved	Reserved	FSDLL_RST_EN	ENVBUSCHG_WKUP	ENIDCHG_WKUP	ENDPDMCHG_WKUP	ENAUTOCLR_PHY_PWD	ENAUTOCLR_CLKGATE	Reserved	WAKEUP_IRQ	ENIRQWAKEUP
Reset	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ENUTMILEVEL3	ENUTMILEVEL2	DATA_ON_LRADC	DEVPLUGIN_IRQ	ENIRQDEVPLUGIN	RESUME_IRQ	ENIRQRESUMEDETECT	RESUMEIRQSTICKY	ENOTGIDDETECT	OTG_ID_CHG_IRQ	DEVPLUGIN_POLARITY	ENDEVPLUGINDETECT	HOSTDISCONDETECT_IRQ	ENIRQHOSTDISCON	ENHOSTDISCONDETECT	ENOTG_ID_CHG_IRQ
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

USBPHYx_CTRLn field descriptions

Field	Description
31 SFTRST	Writing a 1 to this bit will soft-reset the USBPHY_PWD, USBPHY_TX, USBPHY_RX, and USBPHY_CTRL registers.
30 CLKGATE	Gate UTMI Clocks. Clear to 0 to run clocks. Set to 1 to gate clocks. Set this to save power while the USB is not actively being used. Configuration state is kept while the clock is gated. Note this bit can be auto-cleared if there is any wakeup event when USB is suspended while ENAUTOCLR_CLKGATE bit of USBPHY_CTRL is enabled.
29 UTMI_SUSPENDM	Used by the PHY to indicate a powered-down state. If all the power-down bits in the USBPHY_PWD are enabled, UTMI_SUSPENDM will be 0, otherwise 1. UTMI_SUSPENDM is negative logic, as required by the UTMI specification.
28 HOST_FORCE_LS_SE0	Forces the next FS packet that is transmitted to have a EOP with LS timing. This bit is used in host mode for the resume sequence. After the packet is transferred, this bit is cleared. The design can use this function to force the LS SE0 or use the USBPHY_CTRL_UTMI_SUSPENDM to trigger this event when leaving suspend. This bit is used in conjunction with USBPHY_DEBUG_HOST_RESUME_DEBUG.
27 OTG_ID_VALUE	Almost same as OTGID_STATUS in USBPHY_STATUS Register. The only difference is that OTG_ID_VALUE has debounce logic to filter the glitches on ID Pad.

Table continues on the next page...

USBPHY*x*_CTRL*n* field descriptions (continued)

Field	Description
26 Reserved	This field is reserved.
25 Reserved	This field is reserved.
24 FSDLL_RST_EN	Enables the feature to reset the FSDLL lock detection logic at the end of each TX packet.
23 ENVBUSCHG_WKUP	Enables the feature to wakeup USB if VBUS is toggled when USB is suspended.
22 ENIDCHG_WKUP	Enables the feature to wakeup USB if ID is toggled when USB is suspended.
21 ENDPDMCHG_WKUP	Enables the feature to wakeup USB if DP/DM is toggled when USB is suspended. This bit is enabled by default.
20 ENAUTOCLR_PHY_PWD	Enables the feature to auto-clear the PWD register bits in USBPHY_PWD if there is wakeup event while USB is suspended. This should be enabled if needs to support auto wakeup without S/W's interaction.
19 ENAUTOCLR_CLKGATE	Enables the feature to auto-clear the CLKGATE bit if there is wakeup event while USB is suspended. This should be enabled if needs to support auto wakeup without S/W's interaction.
18 Reserved	This field is reserved.
17 WAKEUP_IRQ	Indicates that there is a wakeup event. Reset this bit by writing a 1 to the SCT clear address space and not by a general write.
16 ENIRQWAKEUP	Enables interrupt for the wakeup events.
15 ENUTMILEVEL3	Enables UTMI+ Level3. This should be enabled if needs to support external FS Hub with LS device connected
14 ENUTMILEVEL2	Enables UTMI+ Level2. This should be enabled if needs to support LS device
13 DATA_ON_LRADC	Enables the LRADC to monitor USB_DP and USB_DM. This is for use in non-USB modes only.
12 DEVPLUGIN_IRQ	Indicates that the device is connected. Reset this bit by writing a 1 to the SCT clear address space and not by a general write.
11 ENIRQDEVPLUGIN	Enables interrupt for the detection of connectivity to the USB line.
10 RESUME_IRQ	Indicates that the host is sending a wake-up after suspend. This bit is also set on a reset during suspend. Use this bit to wake up from suspend for either the resume or the reset case. Reset this bit by writing a 1 to the SCT clear address space and not by a general write.
9 ENIRQRESUMEDETECT	Enables interrupt for detection of a non-J state on the USB line. This should only be enabled after the device has entered suspend mode.
8 RESUMEIRQSTICKY	Set to 1 will make RESUME_IRQ bit a sticky bit until software clear it. Set to 0, RESUME_IRQ only set during the wake-up period.
7 ENOTGIDDETECT	Enables circuit to detect resistance of MiniAB ID pin.

Table continues on the next page...

USBPHY*x*_CTRL*n* field descriptions (continued)

Field	Description
6 OTG_ID_CHG_IRQ	OTG ID change interrupt. Indicates the value of ID pin changed.
5 DEVPLUGIN_POLARITY	For device mode, if this bit is cleared to 0, then it trips the interrupt if the device is plugged in. If set to 1, then it trips the interrupt if the device is unplugged.
4 ENDEVPLUGINDETECT	For device mode, enables 200-KOhm pullups for detecting connectivity to the host.
3 HOSTDISCONDETECT_ IRQ	Indicates that the device has disconnected in high-speed mode. Reset this bit by writing a 1 to the SCT clear address space and not by a general write.
2 ENIRQHOSTDISCON	Enables interrupt for detection of disconnection to Device when in high-speed host mode. This should be enabled after ENDEVPLUGINDETECT is enabled.
1 ENHOSTDISCONDETECT	For host mode, enables high-speed disconnect detector. This signal allows the override of enabling the detection that is normally done in the UTMI controller. The UTMI controller enables this circuit whenever the host sends a start-of-frame packet.
0 ENOTG_ID_CHG_IRQ	Enable OTG_ID_CHG_IRQ.

45.2.5 USB PHY Status Register (USBPHYx_STATUS)

The USB PHY Status Register holds results of IRQ and other detects.

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD4					RESUME_STATUS	RSVD3	OTGID_STATUS	RSVD2	DEVPLUGIN_STATUS	RSVD1		HOSTDISCONDETECT_STATUS	RSVD0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBPHYx_STATUS field descriptions

Field	Description
31–11 RSVD4	Reserved.
10 RESUME_STATUS	Indicates that the host is sending a wake-up after suspend and has triggered an interrupt.
9 RSVD3	Reserved.
8 OTGID_STATUS	Indicates the results of ID pin on MiniAB plug. False (0) is when ID resistance is less than Ra_Plug_ID, indicating host (A) side. True (1) is when ID resistance is greater than Rb_Plug_ID, indicating device (B) side.
7 RSVD2	Reserved.
6 DEVPLUGIN_STATUS	Indicates that the device has been connected on the USB_DP and USB_DM lines.
5–4 RSVD1	Reserved.
3 HOSTDISCONDETECT_ STATUS	Indicates that the device has disconnected while in high-speed host mode.
2–0 RSVD0	Reserved.

45.2.6 USB PHY Debug Register (USBPHYx_DEBUG*n*)

This register is used to debug the USB PHY.

Address: Base address + 50h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD3	CLKGATE	HOST_RESUME_DEBUG	SQUELCHRESETLENGTH				ENSQUELCHRESET	RSVD2			SQUELCHRESETCOUNT				
W																
Reset	0	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD1			ENTX2RXCOUNT	TX2RXCOUNT				RSVD0		ENHSTPULLDOWN		HSTPULLDOWN		DEBUG_INTERFACE_HOLD	OTGIDPIOLOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBPHYx_DEBUGn field descriptions

Field	Description
31 RSVD3	Reserved.
30 CLKGATE	Gate Test Clocks. Clear to 0 for running clocks. Set to 1 to gate clocks. Set this to save power while the USB is not actively being used. Configuration state is kept while the clock is gated.
29 HOST_RESUME_DEBUG	Choose to trigger the host resume SE0 with HOST_FORCE_LS_SE0 = 0 or UTMI_SUSPEND = 1.
28–25 SQUELCHRESETLENGTH	Duration of RESET in terms of the number of 480-MHz cycles.
24 ENSQUELCHRESET	Set bit to allow squelch to reset high-speed receive.
23–21 RSVD2	Reserved.
20–16 SQUELCHRESETCOUNT	Delay in between the detection of squelch to the reset of high-speed RX.
15–13 RSVD1	Reserved.
12 ENTX2RXCOUNT	Set this bit to allow a countdown to transition in between TX and RX.
11–8 TX2RXCOUNT	Delay in between the end of transmit to the beginning of receive. This is a Johnson count value and thus will count to 8.
7–6 RSVD0	Reserved.
5–4 ENHSTPULLDOWN	Set bit 5 to 1 to override the control of the USB_DP 15-KOhm pulldown. Set bit 4 to 1 to override the control of the USB_DM 15-KOhm pulldown. Clear to 0 to disable.
3–2 HSTPULLDOWN	Set bit 3 to 1 to pull down 15-KOhm on USB_DP line. Set bit 2 to 1 to pull down 15-KOhm on USB_DM line. Clear to 0 to disable.
1 DEBUG_INTERFACE_ HOLD	Use holding registers to assist in timing for external UTMI interface.
0 OTGIDPIOLOCK	Once OTG ID from USBPHY_STATUS_OTGID_STATUS is sampled, use this to hold the value. This is to save power for the comparators that are used to determine the ID status.

45.2.7 UTMI Debug Status Register 0 (USBPHY*x*_DEBUG0_STATUS)

The UTMI Debug Status Register 0 holds multiple views for counters and status of state machines. This is used in conjunction with the USBPHY_DEBUG1.DBG_ADDRESS field to choose which function to view. The default is described in the bit fields below and is used to count errors.

Address: Base address + 60h offset

Bit	31–26	25–16	15–0
R/W	SQUELCH_COUNT	UTMI_RXERROR_FAIL_COUNT	LOOP_BACK_FAIL_COUNT
Reset	0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

USBPHYx_DEBUG0_STATUS field descriptions

Field	Description
31–26 SQUELCH_ COUNT	Running count of the squelch reset instead of normal end for HS RX.
25–16 UTMI_ RXERROR_ FAIL_COUNT	Running count of the UTMI_RXERROR.
15–0 LOOP_BACK_ FAIL_COUNT	Running count of the failed pseudo-random generator loopback. Each time entering testmode, counter goes to 900D and will count up for every detected packet failure in digital/analog loopback tests.

45.2.8 UTMI Debug Status Register 1 (USBPHY*x*_DEBUG1*n*)

Chooses the muxing of the debug register to be shown in USBPHY_DEBUG0_STATUS.

Address: Base address + 70h offset + (4d × i), where i=0d to 3d

Bit	31–15	14–13	12–0
R	RSVD1	ENTAILADJVD	RSVD0
W			
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0	1 0 0 0 0 0 0 0 0 0 0 0 0

USBPHYx_DEBUG1n field descriptions

Field	Description
31–15 RSVD1	Reserved.
14–13 ENTAILADJVD	Delay increment of the rise of squelch: 00 = Delay is nominal 01 = Delay is +20% 10 = Delay is -20% 11 = Delay is -40%
12–0 RSVD0	Reserved. **Note:** This bit should remain clear.

45.2.9 UTMI RTL Version (USBPHYx_VERSION)

Fields for RTL Version.

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								STEP															
W																																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBPHYx_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
15–0 STEP	Fixed read-only value reflecting the stepping of the RTL version.

45.2.10 USB PHY IP Block Register (USBPHY*x*_IP*n*)

Address: Base address + 90h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	RSVD1													Reserved	Reserved	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD0													EN_USB_CLKS	PLL_LOCKED	PLL_POWER
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBPHY*x*_IP*n* field descriptions

Field	Description
31–19 RSVD1	Reserved.
18 Reserved	Setting this bit will not guarantee chip functionality. This field is reserved.
17 Reserved	Setting this bit will not guarantee chip functionality. This field is reserved.
16 Reserved	Setting this bit will not guarantee chip functionality. This field is reserved.
15–3 RSVD0	Reserved.

Table continues on the next page...

USBPHYx_IPn field descriptions (continued)

Field	Description
2 EN_USB_CLKS	If set to 0, 9-phase PLL outputs for USB PHY are powered down. If set to 1, 9-phase PLL outputs for USB PHY are powered up. Additionally, the UTMICLK120_GATE and UTMICLK30_GATE must be deasserted in the UTMI phy to enable USB operation. This bit came from the clkctrl PIO control block (clkctrl_pllctrl0_en_usb_clks).
1 PLL_LOCKED	Software controlled bit to indicate when the USB PLL has locked. Software needs to wait 10 us after enabling the PLL POWER bit (0) before asserting this bit. If set to 0, tells the UTMI module that the USB PLL has not locked. If set to 1, tells the UTMI module that the USB PLL has locked. Software should clear this bit prior to turning off the USB PLL. This bit came from the clkctrl module.
0 PLL_POWER	USB PLL Power On (0 = PLL off; 1 = PLL On). Allow 10 us after turning the PLL on before using the PLL as a clock source. This is the time the PLL takes to lock to 480 MHz. This bit came from the clkctrl PIO control block (clkctrl_pllctrl0_power).

45.3 USB Analog Memory Map/Register Definition

USB_ANALOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_01A0	USB0 VBUS Detect control register (USB_ANALOG_USB0_VBUS_DETECT)	32	R/W	0010_0004h	45.3.1/2391
4005_01B0	USB0 Charger Detect control register (USB_ANALOG_USB0_CHRG_DETECT)	32	R/W	0009_0000h	45.3.2/2393
4005_01C0	USB0 VBUS Detect Status definition register (USB_ANALOG_USB0_VBUS_DETECT_STATUS)	32	R/W	0000_0000h	45.3.3/2395
4005_01D0	USB0 Charger Detect Status definition register (USB_ANALOG_USB0_CHRG_DETECT_STATUS)	32	R/W	See section	45.3.4/2397
4005_01E0	USB0 Loopback register (USB_ANALOG_USB0_LOOPBACK)	32	R/W	See section	45.3.5/2399
4005_01F0	USB0 Miscellaneous definition register (USB_ANALOG_USB0_MISC)	32	R/W	0000_0002h	45.3.6/2401
4005_0200	USB1 VBUS Detect control register (USB_ANALOG_USB1_VBUS_DETECT)	32	R/W	0010_0004h	45.3.7/2402
4005_0210	USB1 Charger Detect control register (USB_ANALOG_USB1_CHRG_DETECT)	32	R/W	0009_0000h	45.3.8/2405
4005_0220	USB1 VBUS Detect STS definition register (USB_ANALOG_USB1_VBUS_DETECT_STATUS)	32	R	0000_0000h	45.3.9/2407
4005_0230	USB1 Charger Detect Status definition register (USB_ANALOG_USB1_CHRG_DETECT_STATUS)	32	R	See section	45.3.10/ 2409
4005_0240	USB1 Loopback register (USB_ANALOG_USB1_LOOPBACK)	32	R/W	See section	45.3.11/ 2411
4005_0250	USB1 Miscellaneous definition register (USB_ANALOG_USB1_MISC)	32	R/W	0000_0002h	45.3.12/ 2413

45.3.1 USB0 V_{BUS} Detect control register (USB_ANALOG_USB0_VBUS_DETECT)

This register defines the control bits for V_{BUS} detector of USB0.

Address: 4005_0000h base + 1A0h offset = 4005_01A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EN_CHARGER_RESISTOR	0			CHARGE_VBUS	DISCHARGE_VBUS	0					VBUSVALID_PWRUP_CMPS	0	VBUSVALID_TO_B	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								VBUSVALID_OVERRIDE	AVALID_OVERRIDE	BVALID_OVERRIDE	SESSEND_OVERRIDE	VBUS_OVERRIDE_EN	VBUSVALID_THRESH		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

USB_ANALOG_USB0_VBUS_DETECT field descriptions

Field	Description
31 EN_CHARGER_RESISTOR	Enable 125k pullup on USB_DP and 375k on USB_DM to provide USB_CHARGER functionality for USB. This functionality is a legacy feature held over from USB Battery Charging Specification Revision 1.0 and is incompatible with either the plugged-in detector or Battery Charging Specification Revision 1.2. 0 125k pullup on USB_DP and 375k on USB_DM to provide USB_CHARGER functionality for USB is not enabled 1 Enable 125k pullup on USB_DP and 375k on USB_DM to provide USB_CHARGER functionality for USB
30–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 CHARGE_VBUS	USB OTG charge V_{BUS}. 0 Charging of the V_{BUS} is not enabled 1 Enable charging of the V_{BUS}
26 DISCHARGE_VBUS	USB OTG discharge V_{BUS}. 0 Discharging of the V_{BUS} is not enabled 1 Enable discharging of the V_{BUS}

Table continues on the next page...

USB_ANALOG_USB0_VBUS_DETECT field descriptions (continued)

Field	Description
25–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 VBUSVALID_ PWRUP_CMPS	Powers up comparators for $V_{BUS_}$valid detector. 0 Powering up comparators for $V_{BUS_}$valid detector is not enabled 1 Enable powering up comparators for $V_{BUS_}$valid detector.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 VBUSVALID_ TO_B	This bit muxes the Bvalid comparator to the VBUSVALID comparator and is used for test purposes only. 0 Multiplexing of the Bvalid comparator to the VBUSVALID comparator is not enabled 1 Enable multiplexing of the Bvalid comparator to the VBUSVALID comparator
17–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 VBUSVALID_ OVERRIDE	Override value for the VBUSVALID hardware signal. 0 Value for the VBUSVALID hardware signal is not overrided 1 Override value for the VBUSVALID hardware signal.
6 AVALID_ OVERRIDE	Override value for the AVALID hardware signal. 0 AVALID hardware signal is not overrided 1 Override value for the AVALID hardware signal
5 BVALID_ OVERRIDE	Override value for the BVALID hardware signal. 0 BVALID hardware signal is not overrided 1 Override value for the BVALID hardware signal
4 SESSEND_ OVERRIDE	Override value for the SESSEND hardware signal. 0 SESSEND hardware signal is not overrided 1 Override value for the SESSEND hardware signal
3 VBUS_ OVERRIDE_EN	Enable the override of the VBUSVALID, AVALID, BVALID, and SESSEND signals from the USB OTG PHY with override register bit values. 0 Use hardware generated values 1 Use the software controlled values.
2–0 VBUSVALID_ THRESH	Set the threshold for the VBUSVALID comparator. This comparator is the most accurate method to determine the presence of 5v, and includes hysteresis to minimize the need for software debounce of the detection. This comparator has ~50mV of hysteresis to prevent chattering at the comparator trip point. 000 4.0 V 001 4.1 V 010 4.2 V 011 4.3 V 100 4.4 V (default) 101 4.5 V 110 4.6 V 111 4.7 V

45.3.2 USB0 Charger Detect control register (USB_ANALOG_USB0_CHRG_DETECT)

This register defines the control bits for the USB charger detector of USB0.

Address: 4005_0000h base + 1B0h offset = 4005_01B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0	CHRG_DET_CTRL	CHRG_DET_STATUS	EN_B	CHK_CHRG_B	CHK_CONTACT	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															FORCE_DETECT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_ANALOG_USB0_CHRG_DETECT field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 CHRG_DET_ CTRL	Forces internal USB charging circuitery to enable basic USB battery charging v1.1 device charger detection functionality. 0 USB battery charging v1.1 device charger detection functionality is not enabled 1 Enable basic USB battery charging v1.1 device charger detection functionality
21 CHRG_DET_ STATUS	Status of internal charge detection. 0 Charger not detected. 1 Charger detected.
20 EN_B	This bit enables the charger detector. 0 Enable the charger detector. 1 Disable the charger detector.
19 CHK_CHRG_B	This bit checks whether a charger (either a dedicated charger or a host charger) is connected to USB port. 0 Check whether a charger (either a dedicated charger or a host charger) is connected to USB port. 1 Do not check whether a charger is connected to the USB port.
18 CHK_CONTACT	This bit checks whether the USB plug has been in contact with each other. 0 Do not check the contact of USB plug. 1 Check whether the USB plug has been in contact with each other.
17–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 FORCE_ DETECT	Set this bit to 1 to force the charger detector circuit to signal the presence of a charger. 0 The charger detector circuit is not forced to signal the presence of a charger. 1 Force the charger detector circuit to signal the presence of a charger.

45.3.3 USB0 V_{BUS} Detect Status definition register (USB_ANALOG_USB0_VBUS_DETECT_STATUS)

This register defines the status bits of the V_{BUS} detectors of USB1.

Address: 4005_0000h base + 1C0h offset = 4005_01C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												VBUS_VALID	AVALID	BVALID	SESSEND
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_ANALOG_USB0_VBUS_DETECT_STATUS field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

USB_ANALOG_USB0_VBUS_DETECT_STATUS field descriptions (continued)

Field	Description
3 VBUS_VALID	V_{BUS} valid for USB OTG. This bit is a read only version of the state of the analog signal. It cannot be overwritten by software. 0 V_{BUS} not valid for USB OTG 1 V_{BUS} valid for USB OTG
2 AVALID	Indicates V_{BUS} is valid for a A-peripheral. This bit is a read only version of the state of the analog signal. It cannot be overritten by software. 0 V_{BUS} is not valid for a A-peripheral 1 V_{BUS} is valid for a A-peripheral
1 BVALID	V_{BUS} valid for USB B-session. This bit is a read only version of the state of the analog signal. It cannot be overwritten by software. 0 V_{BUS} is not valid for USB B-session 1 V_{BUS} valid for USB B-session
0 SESSEND	Session end for USB OTG. This bit is a read only version of the state of the analog signal. It cannot be overwritten by software like the SESSEND bit below. Note: This bit's default value depends on whether VDD5V is present. 0 VDD5V is present. 1 VDD5V is not present.

45.3.4 USB0 Charger Detect Status definition register (USB_ANALOG_USB0_CHRG_DETECT_STATUS)

This register defines the status bits for the USB charger detector of USB1.

Address: 4005_0000h base + 1D0h offset = 4005_01D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												DP_STATE	DM_STATE	CHRG_DETECTED	PLUG_CONTACT
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

USB_ANALOG_USB0_CHRG_DETECT_STATUS field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 DP_STATE	DP line state output of the charger detector.
2 DM_STATE	DM line state output of the charger detector.
1 CHRG_ DETECTED	This bit is a read only version of the state of the analog signal. 0 The USB port is not connected to a charger. 1 A charger (either a dedicated charger or a host charger) is connected to the USB port.
0 PLUG_ CONTACT	This bit shows the contact status of the USB plug. 0 The USB plug has not been contacted. 1 The USB plug has made good contact.

45.3.5 USB0 Loopback register (USB_ANALOG_USB0_LOOPBACK)

This register defines the status bits for the USB charger detector.

Address: 4005_0000h base + 1E0h offset = 4005_01E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							UTMO_DIG_TST1	UTMO_DIG_TST0	TSTI_TX_HIZ	TSTI_TX_EN	TSTI_TX_LS_MODE	TSTI_TX_HS_MODE	UTMI_DIG_TST1	UTMI_DIG_TST0	UTMI_TESTSTART
W																
Reset	0	0	0	0	0	0	0	x*	x*	0	0	0	0	0	0	0

* Notes:

- x = Undefined at reset.

USB_ANALOG_USB0_LOOPBACK field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 UTMO_DIG_ TST1	This read-only bit is a status bit for USB0 Loopback. 0 Pass 1 Not pass
7 UTMO_DIG_ TST0	This read-only bit is a status bit for USB0 Loopback. 0 Not pass 1 Pass
6 TSTI_TX_HIZ	Makes TX HIZ for USB0. 0 TX is not HIZ for USB0. 1 Make TX HIZ for USB0.
5 TSTI_TX_EN	Enables TX for USB0. 0 TX for USB0 is not enabled. 1 Enable TX for USB0.
4 TSTI_TX_LS_ MODE	Chooses LS mode for USB0. 0 Choose HS or FS mode which is defined by TSTI1_TX_HS. 1 Choose LS for USB0.
3 TSTI_TX_HS_ MODE	Chooses HS or FS mode for USB0. 0 USB0 FS mode. 1 USB0 HS mode.
2 UTMI_DIG_TST1	Test 1 loopback mode for USB0. 0 USB0 loopback test 1 is not enabled. 1 Enable USB0 loopback test 1.
1 UTMI_DIG_TST0	Test 0 loopback mode for USB0. 0 USB0 loopback test 0 is not enabled. 1 Enable USB0 loopback test 0.
0 UTMI_ TESTSTART	Enables the USB0 loopback test. 0 USB0 loopback test is not enabled. 1 Enable USB0 loopback test.

45.3.6 USB0 Miscellaneous definition register (USB_ANALOG_USB0_MISC)

Address: 4005_0000h base + 1F0h offset = 4005_01F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	EN_CLK_TO_UTMI	0	0	0	0	0	0	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														EN_DEGLITCH	HS_USE_ EXTERNAL_R
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

USB_ANALOG_USB0_MISC field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 EN_CLK_TO_UTMI	Enables the clk to the UTMI block.
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 EN_DEGLITCH	Enables the deglitching circuit of the USB PLL output.

Table continues on the next page...

USB_ANALOG_USB0_MISC field descriptions (continued)

Field	Description
0 HS_USE_ EXTERNAL_R	Use external resistor to generate the current bias for the high speed transmitter. This bit should not be changed unless recommended by Freescale.

45.3.7 USB1 V_{BUS} Detect control register (USB_ANALOG_USB1_VBUS_DETECT)

This register defines the control bits for V_{BUS} detector of USB1.

Address: 4005_0000h base + 200h offset = 4005_0200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EN_CHARGER_RESISTOR	0			CHARGE_VBUS	DISCHARGE_VBUS	0					VBUSVALID_PWRUP_CMPS	0	VBUSVALID_TO_B	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								VBUSVALID_OVERRIDE	AVALID_OVERRIDE	BVALID_OVERRIDE	SESSEND_OVERRIDE	VBUS_OVERRIDE_EN	VBUSVALID_THRESH		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

USB_ANALOG_USB1_VBUS_DETECT field descriptions

Field	Description
31 EN_CHARGER_ RESISTOR	Enable 125k pullup on USB_DP and 375k on USB_DM to provide USB_CHARGER functionality for USB. This functionality is a legacy feature held over from USB Battery Charging Specification Revision 1.0 and is incompatible with either the plugged-in detector or Battery Charging Specification Revision 1.2. 0 125k pullup on USB_DP and 375k on USB_DM to provide USB_CHARGER functionality for USB is not enabled 1 Enable 125k pullup on USB_DP and 375k on USB_DM to provide USB_CHARGER functionality for USB
30–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

USB_ANALOG_USB1_VBUS_DETECT field descriptions (continued)

Field	Description
27 CHARGE_VBUS	USB OTG charge V_{BUS}. 0 Charging of the V_{BUS} is not enabled 1 Enable charging of the V_{BUS}
26 DISCHARGE_ VBUS	USB OTG discharge V_{BUS}. 0 Discharging of the V_{BUS} is not enabled 1 Enable discharging of the V_{BUS}
25–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 VBUSVALID_ PWRUP_CMPS	Powers up comparators for $V_{BUS_}$valid detector. 0 Powering up comparators for $V_{BUS_}$valid detector is not enabled 1 Enable powering up comparators for $V_{BUS_}$valid detector.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 VBUSVALID_ TO_B	This bit muxes the Bvalid comparator to the VBUSVALID comparator and is used for test purposes only. 0 Multiplexing of the Bvalid comparator to the VBUSVALID comparator is not enabled 1 Enable multiplexing of the Bvalid comparator to the VBUSVALID comparator
17–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 VBUSVALID_ OVERRIDE	Override value for the VBUSVALID hardware signal. 0 Value for the VBUSVALID hardware signal is not overrided 1 Override value for the VBUSVALID hardware signal.
6 AVALID_ OVERRIDE	Override value for the AVALID hardware signal. 0 AVALID hardware signal is not overrided 1 Override value for the AVALID hardware signal
5 BVALID_ OVERRIDE	Override value for the BVALID hardware signal. 0 BVALID hardware signal is not overrided 1 Override value for the BVALID hardware signal
4 SESSEND_ OVERRIDE	Override value for the SESSEND hardware signal. 0 SESSEND hardware signal is not overrided 1 Override value for the SESSEND hardware signal
3 VBUS_ OVERRIDE_EN	Enable the override of the VBUSVALID, AVALID, BVALID, and SESSEND signals from the USB OTG PHY with override register bit values. 0 Use hardware generated values 1 Use the software controlled values.
2–0 VBUSVALID_ THRESH	Set the threshold for the VBUSVALID comparator. This comparator is the most accurate method to determine the presence of 5v, and includes hystersis to minimize the need for software debounce of the detection. This comparator has ~50mV of hystersis to prevent chattering at the comparator trip point. 000 4.0 V 001 4.1 V

Table continues on the next page...

USB_ANALOG_USB1_VBUS_DETECT field descriptions (continued)

Field	Description
	010 4.2 V 011 4.3 V 100 4.4 V (default) 101 4.5 V 110 4.6 V 111 4.7 V

45.3.8 USB1 Charger Detect control register (USB_ANALOG_USB1_CHRG_DETECT)

This register defines the control bits for the USB charger detector of USB1.

Address: 4005_0000h base + 210h offset = 4005_0210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0	CHRG_DET_CTRL	CHRG_DET_STATUS	EN_B	CHK_CHRG_B	CHK_CONTACT	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															FORCE_DETECT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_ANALOG_USB1_CHRG_DETECT field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 CHRG_DET_ CTRL	Forces internal USB charging circuitery to enable basic USB battery charging v1.1 device charger detection functionality. 0 USB battery charging v1.1 device charger detection functionality is not enabled 1 Enable basic USB battery charging v1.1 device charger detection functionality
21 CHRG_DET_ STATUS	Status of internal charge detection. 0 Charger not detected. 1 Charger detected.
20 EN_B	This bit enables the charger detector. 0 Enable the charger detector. 1 Disable the charger detector.
19 CHK_CHRG_B	This bit checks whether a charger (either a dedicated charger or a host charger) is connected to USB port. 0 Check whether a charger (either a dedicated charger or a host charger) is connected to USB port. 1 Do not check whether a charger is connected to the USB port.
18 CHK_CONTACT	This bit checks whether the USB plug has been in contact with each other. 0 Do not check the contact of USB plug. 1 Check whether the USB plug has been in contact with each other.
17–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 FORCE_ DETECT	Set this bit to 1 to force the charger detector circuit to signal the presence of a charger. 0 The charger detector circuit is not forced to signal the presence of a charger. 1 Force the charger detector circuit to signal the presence of a charger.

45.3.9 USB1 V_{BUS} Detect STS definition register (USB_ANALOG_USB1_VBUS_DETECT_STATUS)

This register defines the status bits of the V_{BUS} detectors of USB1.

Address: 4005_0000h base + 220h offset = 4005_0220h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												VBUS_VALID	AVALID	BVALID	SESSEND
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USB_ANALOG_USB1_VBUS_DETECT_STATUS field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

USB_ANALOG_USB1_VBUS_DETECT_STATUS field descriptions (continued)

Field	Description
3 VBUS_VALID	V_{BUS} valid for USB OTG. This bit is a read only version of the state of the analog signal. It cannot be overwritten by software. 0 V_{BUS} not valid for USB OTG 1 V_{BUS} valid for USB OTG
2 AVALID	Indicates V_{BUS} is valid for a A-peripheral. This bit is a read only version of the state of the analog signal. It cannot be overritten by software. 0 V_{BUS} is not valid for a A-peripheral 1 V_{BUS} is valid for a A-peripheral
1 BVALID	V_{BUS} valid for B-session. This bit is a read only version of the state of the analog signal. It cannot be overwritten by software. 0 V_{BUS} is not valid for USB B-session 1 V_{BUS} valid for USB B-session
0 SESSEND	Session end for USB OTG. This bit is a read only version of the state of the analog signal. It cannot be overwritten by software like the SESSEND bit below. Note: This bit's default value depends on whether VDD5V is present. 0 VDD5V is present. 1 VDD5V is not present.

45.3.10 USB1 Charger Detect Status definition register (USB_ANALOG_USB1_CHRG_DETECT_STATUS)

This register defines the status bits for the USB charger detector of USB1.

Address: 4005_0000h base + 230h offset = 4005_0230h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												DP_STATE	DM_STATE	CHRG_DETECTED	PLUG_CONTACT
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

USB_ANALOG_USB1_CHRG_DETECT_STATUS field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 DP_STATE	DP line state output of the charger detector.
2 DM_STATE	DM line state output of the charger detector.
1 CHRG_ DETECTED	This bit is a read only version of the state of the analog signal. 0 The USB port is not connected to a charger. 1 A charger (either a dedicated charger or a host charger) is connected to the USB port.
0 PLUG_ CONTACT	This bit shows the contact status of the USB plug. 0 The USB plug has not been contacted. 1 The USB plug has made good contact.

45.3.11 USB1 Loopback register (USB_ANALOG_USB1_LOOPBACK)

This register defines the status bits for the USB1 charger detector.

Address: 4005_0000h base + 240h offset = 4005_0240h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							UTM1_DIG_TST1	UTM1_DIG_TST0	TSTI_TX_HIZ	TSTI_TX_EN	TSTI_TX_LS_MODE	TSTI_TX_HS_MODE	UTMI_DIG_TST1	UTMI_DIG_TST0	UTMI_TESTSTART
W																
Reset	0	0	0	0	0	0	0	x*	x*	0	0	0	0	0	0	0

* Notes:
- x = Undefined at reset.

USB_ANALOG_USB1_LOOPBACK field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 UTM1_DIG_ TST1	This read-only bit is a status bit for USB0 Loopback. 0 Pass 1 Not pass
7 UTM1_DIG_ TST0	This read-only bit is a status bit for USB1 Loopback. 0 Not pass 1 Pass
6 TSTI_TX_HIZ	Makes TX HIZ for USB1. 0 TX is not HIZ for USB1. 1 Make TX HIZ for USB1.
5 TSTI_TX_EN	Enables TX for USB1. 0 TX for USB1 is not enabled. 1 Enable TX for USB1.
4 TSTI_TX_LS_ MODE	Chooses LS mode for USB1. 0 Choose HS or FS mode which is defined by TSTI1_TX_HS. 1 Choose LS for USB1.
3 TSTI_TX_HS_ MODE	Chooses HS or FS mode for USB1. 0 USB1 FS mode. 1 USB1 HS mode.
2 UTMI_DIG_TST1	Test 1 loopback mode for USB1. 0 USB1 loopback test 1 is not enabled. 1 Enable USB1 loopback test 1.
1 UTMI_DIG_TST0	Test 0 loopback mode for USB1. 0 USB1 loopback test 0 is not enabled. 1 Enable USB1 loopback test 0.
0 UTMI_ TESTSTART	Enables the USB1 loopback test. 0 USB1 loopback test is not enabled. 1 Enable USB1 loopback test.

45.3.12 USB1 Miscellaneous definition register (USB_ANALOG_USB1_MISC)

Address: 4005_0000h base + 250h offset = 4005_0250h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	EN_CLK_TO_UTMI	0	0	0	0	0	0	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														EN_DEGLITCH	HS_USE_EXTERNAL_R
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

USB_ANALOG_USB1_MISC field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 EN_CLK_TO_UTMI	Enables the clk to the UTMI block.
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 EN_DEGLITCH	Enables the deglitching circuit of the USB PLL output.

Table continues on the next page...

USB_ANALOG_USB1_MISC field descriptions (continued)

Field	Description
0 HS_USE_ EXTERNAL_R	Use external resistor to generate the current bias for the high speed transmitter. This bit should not be changed unless recommended by Freescale.

45.4 Operation

The UTM provides a 16-bit interface to the USB controller. This interface is clocked at 30 MHz.

- The digital portions of the USBPHY block include the UTMI, digital transmitter, digital receiver, and the programmable registers.
- The analog transceiver section comprises an analog receiver and an analog transmitter, as shown in Figure 45-116.

45.4.1 UTMI

The UTMI block handles the line_state bits, reset buffering, suspend distribution, transceiver speed selection, and transceiver termination selection.

The PLL supplies a 120 MHz signal to all of the digital logic. The UTMI block does a final divide-by-four to develop the 30 MHz clock used in the interface.

45.4.2 Digital Transmitter

The digital transmitter receives the 16-bit transmit data from the USB controller and handles the tx_valid, tx_validh and tx_ready handshake.

In addition, it contains the transmit serializer that converts the 16-bit parallel words at 30 MHz to a single bitstream at 480 Mbit for high-speed or 12 Mbit for full-speed or 1.5 Mbit for low-speed. It does this while implementing the bit-stuffing algorithm and the NRZI encoder that are used to remove the DC component from the serial bitstream. The output of this encoder is sent to the low-speed (LS), full-speed (FS) or high-speed (HS) drivers in the analog transceiver section's transmitter block.

45.4.3 Digital Receiver

The digital receiver receives the raw serial bitstream from the low speed (LS) differential transceiver, full speed (FS) differential transceiver, and a 9X, 480 MHz sampled data from the high speed (HS) differential transceiver.

As the phase of the USB host transmitter shifts relative to the local PLL, the receiver section's HS DLL tracks these changes to give a reliable sample of the incoming 480 Mbit/s bitstream. Since this sample point shifts relative to the PLL phase used by the digital logic, a rate-matching elastic buffer is provided to cross this clock domain boundary. Once the bitstream is in the local clock domain, an NRZI decoder and bit unstuffer restore the original payload data bitstream and pass it to a deserializer and holding register. The receive state machine handles the rx_valid, rx_validh, and handshake with the USB controller. The handshake is not interlocked, in that there is no rx_ready signal coming from the controller. The controller must take each 16-bit value as presented by the PHY. The receive state machine provides an rx_active signal to the controller that indicates when it is inside a valid packet (SYNC detected, and so on).

45.4.4 Analog Receiver

The analog receiver comprises five differential receivers, two single-ended receivers, and a 9X, 480 MHz HS data sampling module

, as shown in the figure below and described further in this section.

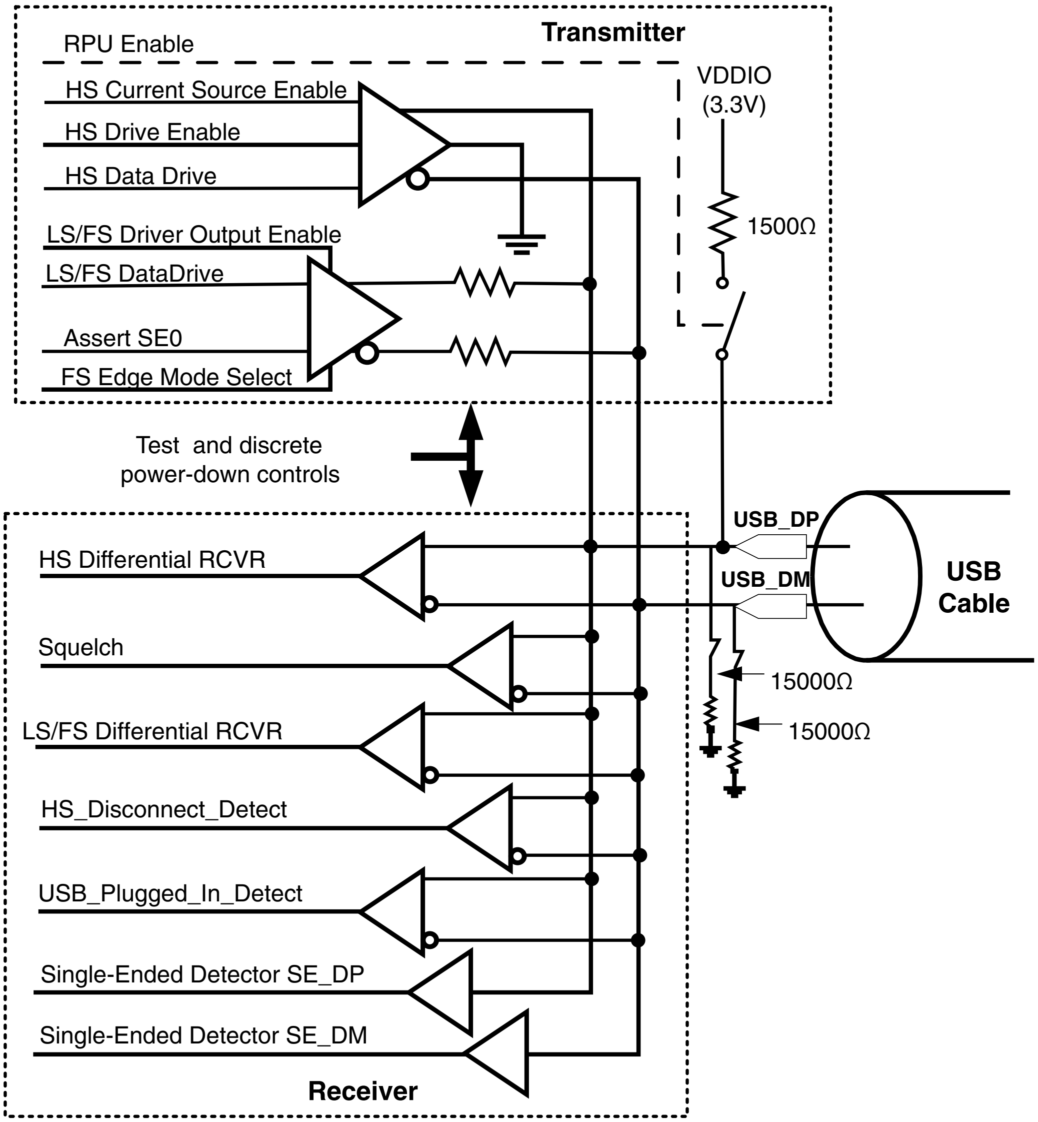


Figure 45-116. USB 2.0 PHY Analog Transceiver Block Diagram

45.4.4.1 HS Differential Receiver

The high-speed differential receiver is both a differential analog receiver and threshold comparator. Its output is a one if the differential signal is greater than a 0-V threshold.

Otherwise, its output is 0. Its purpose is to discriminate the ± 400-mV differential voltage resulting from the high-speed drivers current flow into the dual 45Ω terminations found on each leg of the differential pair. The envelope or squelch detector, described below, ensures that the differential signal has sufficient magnitude to be valid. The HS differential receiver tolerates up to 500 mV of common mode offset.

45.4.4.2 Squelch Detector

The squelch detector is a differential analog receiver and threshold comparator.

Its output is 1, if the differential magnitude is less than a nominal 100 mV threshold. Otherwise, its output is 0.

Its purpose is to invalidate the HS differential receiver when the incoming signal is simply too low to receive reliably.

45.4.4.3 LS/FS Differential Receiver

The low-speed/full-speed differential receiver is both a differential analog receiver and threshold comparator.

The crossover voltage falls between 1.3 V and 2.0 V. Its output is 1, when the USB_DP line is above the crossover point and the USB_DM line is below the crossover point. The digital receiver section decodes the receiver data into J or K state according to the speed.

45.4.4.4 HS Disconnect Detector

This host-side function is not used in applications, but is included to make a complete UTMI macrocell. It is a differential analog receiver and threshold comparator.

It outputs high when differential magnitude is greater than a nominal 575-mV threshold. Otherwise, it outputs low.

45.4.4.5 USB Plugged-In Detector

The USB plugged-in detector looks for both USB_DP and USB_DM to be high. There is a pair of large on-chip pullup resistors (200 KΩ) that hold both USB_DP and USB_DM high when the USB cable is not attached. The USB plugged-in detector signals a 0 in this case.

When operating in device mode, the upstream port in host/hub interface contains a 15 KΩ pulldown resistor which could easily override the 200 KΩ pullup resistor. When plugged in, at least one signal in the pair will be low, which will force the plugged-in detector's output high.

45.4.4.6 Single-Ended USB_DP Receiver

The single-ended USB_DP receiver output is high whenever the USB_DP input is above its nominal 1.8 V threshold.

45.4.4.7 Single-Ended USB_DM Receiver

The single-ended USB_DM receiver output is high whenever the USB_DM input is above its nominal 1.8 V threshold.

45.4.4.8 9X Oversample Module

The 9X oversample module uses nine identically spaced phases of the 480 MHz clock to sample a high speed bit data. The squelch signal is sampled only 1X.

45.4.5 Analog Transmitter

The analog transmitter comprises two differential drivers: one for high-speed signaling and one for full-speed signaling. It also contains the switchable 1.5 KΩ pullup resistor.

See Figure 1.

45.4.5.1 Switchable High-Speed 45Ω Termination Resistors

High-speed current mode differential signaling requires good 90 Ω differential termination at each end of the USB cable. This results from switching in 45 Ω terminating resistors from each signal line to ground at each end of the cable.

Because each signal is parallel terminated with 45 Ω at each end, each driver sees a 22.5 Ω load. This load impedance is much too low for full-speed signaling levels—hence the need for switchable high-speed terminating resistors. Switchable trimming resistors are

provided to tune the actual termination resistance of each device, as shown in Figure 1. The USBPHY_TX_TXCAL45DP bit field, for example, allows one of 16 trimming resistor values to be placed in parallel with the 45Ω terminator on the USB_DP signal.

45.4.5.2 Low-Speed/Full-Speed Differential Driver

The low-speed/full-speed differential drivers are essentially "open drain" low-impedance pulldown devices that are switched in a differential mode for low-speed or full-speed signaling, that is, either one or the other device is turned on to signal the "J" state or the "K" state.

The tx_ls_en signal is used to select the USB_DP/USB_DM edge for low-speed or full-speed. Setting this bit to 1 selects the low-speed driver; otherwise the full-speed driver is selected. These drivers are both turned on, simultaneously, for high-speed signaling. This has the effect of switching in both 45 Ω terminating resistors. The tx_fs_hiz signal originates in the digital transmitter section. The hs_term signal that also controls these drivers comes from the UTMI.

45.4.5.3 High-Speed Differential Driver

The high-speed differential driver receives a 17.78 mA current from the constant current source (Iref) and essentially steers it down either the USB_DP signal or the USB_DM signal or alternatively to ground.

This current will produce approximately a 400 mV drop across the 22.5 Ω termination seen by the driver when it is steered onto one of the signal lines. The approximately 17.78 mA current source is referenced back to the integrated voltage-band-gap (Vbg) circuit. The Iref, Ibias, and V to I circuits are shared with the integrated battery charger.

45.4.5.4 Switchable 1.5KΩ USB_DP Pullup Resistor

This product contains a switchable 1.5 KΩ pullup resistor on the USB DP signal.

This resistor is switched on to indicate to the host/hub controller that a full-speed-capable device is on the USB cable, powered on, and ready. This resistor is switched off at power-on reset so the host does not recognize a USB device until the processor software enables the announcement of a full-speed device.

45.4.5.5 Switchable 15KΩ USB_DP Pulldown Resistor

This product contains a switchable 15 KΩ pulldown resistor on both USB_DP and USB_DM signals. This is used in host mode to indicate to the device controller that a host is present.

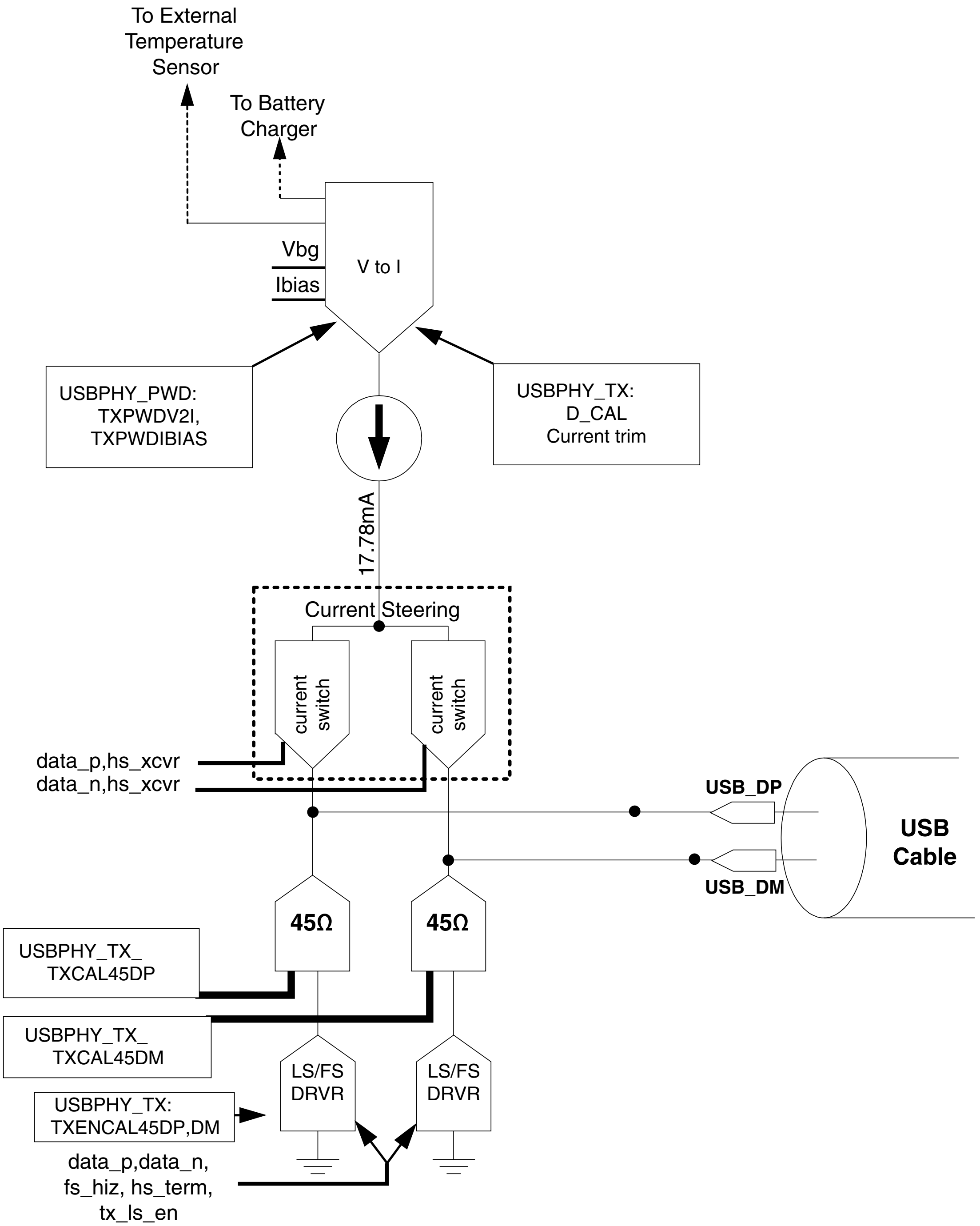


Figure 45-117. USB 2.0 PHY Transmitter Block Diagram

The table below summarizes the response of the PHY analog transmitter to various states of UTMI input and key transmit/receive state machine states.

Table 45-119. USB PHY Terminator States

UTMI OPMODE	UTM TERM	UTM XCVR	T/R	Function	45 Ω HI-Z	1500 Ω HI-Z	
00=Normal	0	00	X	HS	0	1	
	1	01/11	T	FS	0	0	
	1	10	T	LS	0	0	
	1	01/11	R	FS	1	0	SUSPEND
	1	10	R	LS	1	0	SUSPEND
	1	00	R	CHIRP	1	0	
	1	00	T	CHIRP	1	0	
	0	01	X	DISCONNECT	1	1	
01=NoDrive	0	00	T	HS	1	1	
	0	00	R	HS	1	1	
	1	01	X	FS	1	1	
	1	00	X	CHIRP	1	1	
	0	01	X	DISCONNECT	1	1	POR
10=NoNRZI NoBitStuff	0	00	X	HS	0	1	
	1	01	T	FS	0	0	
	1	01	R	FS	1	0	
	1	00	R	CHIRP	1	0	
	1	00	T	CHIRP	1	0	
	0	01	X	DISCONNECT	1	1	
11= Invalid	0	00	T	HS	1	1	
	0	00	R	HS	1	1	
	1	01	X	FS	1	1	
	1	00	X	CHIRP	1	1	
	0	01	X	DISCONNECT	1	1	

45.4.6 Recommended Register Configuration for USB Certification

The register settings in this section are recommended for passing USB certification.

The following settings lower the J/K levels to certifiable limits:

HW_USBPHY_TX_TXCAL45DP = 0x0
HW_USBPHY_TX_TXCAL45DM = 0x0
HW_USBPHY_TX_D_CAL = 0x7

Chapter 46
CAN (FlexCAN)

46.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The FlexCAN module is a communication controller implementing the CAN protocol according to the CAN 2.0 B protocol specification. A general block diagram is shown in the following figure, which describes the main subblocks implemented in the FlexCAN module, including one associated memory for storing message buffers, Receive (Rx) Global Mask registers, Receive Individual Mask registers, Receive FIFO filters, and Receive FIFO ID filters. The functions of the submodules are described in subsequent sections.

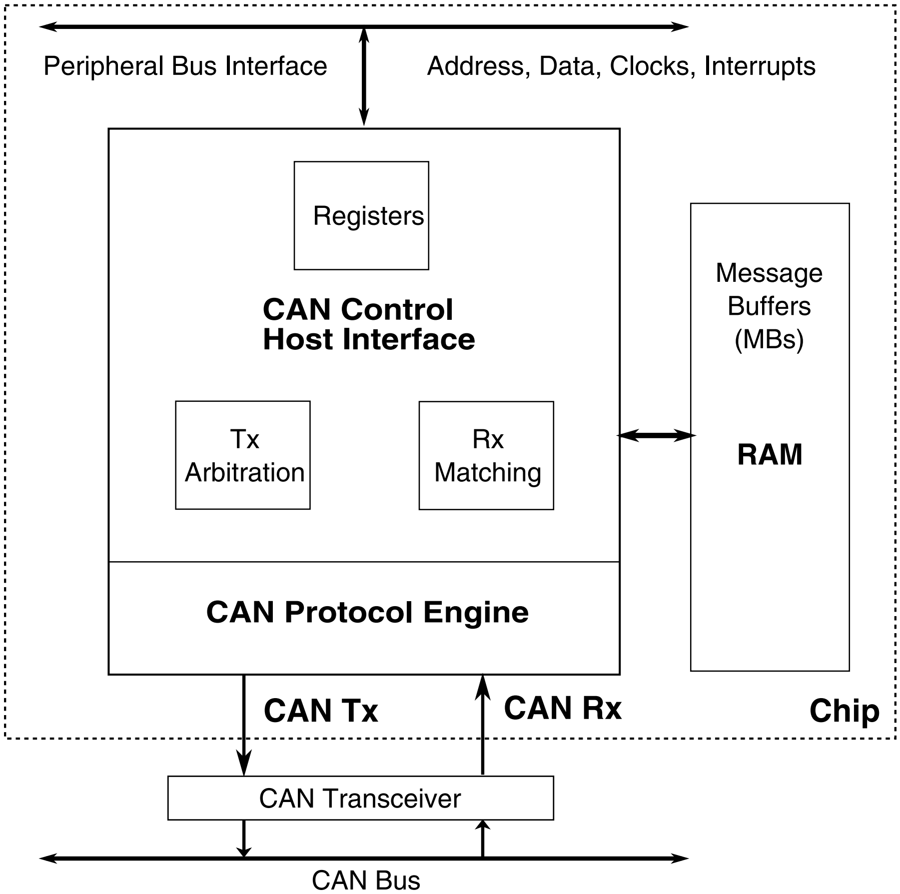


Figure 46-1. FlexCAN block diagram

46.1.1 Overview

The CAN protocol was primarily designed to be used as a vehicle serial data bus, meeting the specific requirements of this field:

- Real-time processing
- Reliable operation in the EMI environment of a vehicle
- Cost-effectiveness
- Required bandwidth

The FlexCAN module is a full implementation of the CAN protocol specification, Version 2.0 B, which supports both standard and extended message frames. The message buffers are stored in an embedded RAM dedicated to the FlexCAN module. See the chip configuration details for the actual number of message buffers configured in the MCU.

The CAN Protocol Engine (PE) submodule manages the serial communication on the CAN bus:

- Requesting RAM access for receiving and transmitting message frames

- Validating received messages
- Performing error handling

The Controller Host Interface (CHI) sub-module handles message buffer selection for reception and transmission, taking care of arbitration and ID matching algorithms.

The Bus Interface Unit (BIU) sub-module controls the access to and from the internal interface bus, in order to establish connection to the CPU and to other blocks. Clocks, address and data buses, interrupt outputs and test signals are accessed through the BIU.

46.1.2 FlexCAN module features

The FlexCAN module includes these distinctive legacy features:

- Full implementation of the CAN protocol specification, Version 2.0 B
    - Standard data and remote frames
    - Extended data and remote frames
    - Zero to eight bytes data length
    - Programmable bit rate up to 1 Mb/sec
    - Content-related addressing
- Flexible mailboxes of zero to eight bytes data length
- Each mailbox configurable as receive or transmit, all supporting standard and extended messages
- Individual Rx Mask registers per mailbox
- Full-featured Rx FIFO with storage capacity for up to six frames and automatic internal pointer handling
- Transmission abort capability
- Programmable clock source to the CAN Protocol Interface, either bus clock or crystal oscillator
- Unused structures space can be used as general purpose RAM space
- Listen-Only mode capability
- Programmable Loop-Back mode supporting self-test operation

- Programmable transmission priority scheme: lowest ID, lowest buffer number, or highest priority
- Time stamp based on 16-bit free-running timer
- Global network time, synchronized by a specific message
- Maskable interrupts
- Independence from the transmission medium (an external transceiver is assumed)
- Short latency time due to an arbitration scheme for high-priority messages
- Low power modes

New major features are also provided:

- Remote request frames may be handled automatically or by software
- CAN bit time settings and configuration bits can only be written in Freeze mode
- Tx mailbox status (Lowest priority buffer or empty buffer)
- Identifier Acceptance Filter Hit Indicator (IDHIT) register for received frames
- SYNCH bit available in Error in Status 1 register to inform that the module is synchronous with CAN bus
- CRC status for transmitted message
- Rx FIFO Global Mask register
- Selectable priority between mailboxes and Rx FIFO during matching process
- Powerful Rx FIFO ID filtering, capable of matching incoming IDs against either 128 extended, 256 standard, or 512 partial (8 bit) IDs, with up to 32 individual masking capability
- 100% backward compatibility with previous FlexCAN version
- Supports detection and correction of errors in memory read accesses. Each byte of FlexCAN memory is associated to 5 parity bits and the error correction mechanism assures that in this 13-bit word, errors in one bit can be corrected (corrected errors) and errors in 2 bits can be detected but not corrected (non-corrected errors).

46.1.3 Modes of operation

The FlexCAN module has these functional modes:

- Normal mode (User or Supervisor):

  In Normal mode, the module operates receiving and/or transmitting message frames, errors are handled normally, and all CAN Protocol functions are enabled. User and Supervisor Modes differ in the access to some restricted control registers.

- Freeze mode:

  Freeze mode is enabled when the FRZ bit in MCR is asserted. If enabled, Freeze mode is entered when MCR[HALT] is set or when Debug mode is requested at MCU level and MCR[FRZ_ACK] is asserted by the FlexCAN. In this mode, no transmission or reception of frames is done and synchronicity to the CAN bus is lost. See Freeze mode for more information.

- Listen-Only mode:

  The module enters this mode when the LOM field in the Control 1 Register is asserted. In this mode, transmission is disabled, all error counters are frozen, and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message.

- Loop-Back mode:

  The module enters this mode when the LPB field in the Control 1 Register is asserted. In this mode, FlexCAN performs an internal loop back that can be used for self-test operation. The bit stream output of the transmitter is internally fed back to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic '1'). FlexCAN behaves as it normally does when transmitting and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

For low-power operation, the FlexCAN module has:

- Module Disable mode:

  This low-power mode is entered when the MDIS bit in the MCR Register is asserted by the CPU and the LPM_ACK is asserted by the FlexCAN. When disabled, the module requests to disable the clocks to the CAN Protocol Engine and Controller Host Interface submodules. Exit from this mode is done by negating the MDIS bit in the MCR register. See Module Disable mode for more information.

- Stop mode:

This low power mode is entered when Stop mode is requested at MCU level and the LPM_ACK bit in the MCR Register is asserted by the FlexCAN. When in Stop Mode, the module puts itself in an inactive state and then informs the CPU that the clocks can be shut down globally. Exit from this mode happens when the Stop mode request is removed. See Stop mode for more information.

46.2 FlexCAN signal descriptions

The FlexCAN module has two I/O signals connected to the external MCU pins. These signals are summarized in the following table and described in more detail in the next subsections.

Table 46-1. FlexCAN signal descriptions

Signal	Description	I/O
CAN Rx	CAN Receive Pin	Input
CAN Tx	CAN Transmit Pin	Output

46.2.1 CAN Rx

This pin is the receive pin from the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

46.2.2 CAN Tx

This pin is the transmit pin to the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

46.3 Memory map/register definition

This section describes the registers and data structures in the FlexCAN module. The base address of the module depends on the particular memory map of the MCU.

46.3.1 FlexCAN memory mapping

The complete memory map for a FlexCAN module is shown in the following table.

The address space occupied by FlexCAN has 128 bytes for registers starting at the module base address, followed by embedded RAM starting at address 0x0080.

Each individual register is identified by its complete name and the corresponding mnemonic. The access type can be Supervisor (S) or Unrestricted (U). Most of the registers can be configured to have either Supervisor or Unrestricted access by programming the SUPV field in the MCR register. These registers are identified as S/U in the Access column of Table 46-2.

Table 46-2. Register access and reset information

Register	Access type	Affected by hard reset	Affected by soft reset
Module Configuration Register (MCR)	S	Yes	Yes
Control 1 register (CTRL1)	S/U	Yes	No
Free Running Timer register (TIMER)	S/U	Yes	Yes
Rx Mailboxes Global Mask register (RXMGMASK)	S/U	No	No
Rx Buffer 14 Mask register (RX14MASK)	S/U	No	No
Rx Buffer 15 Mask register (RX15MASK)	S/U	No	No
Error Counter Register (ECR)	S/U	Yes	Yes
Error and Status 1 Register (ESR1)	S/U	Yes	Yes
Interrupt Masks 2 register (IMASK2)	S/U	Yes	Yes
Interrupt Masks 1 register (IMASK1)	S/U	Yes	Yes
Interrupt Flags 2 register (IFLAG2)	S/U	Yes	Yes
Interrupt Flags 1 register (IFLAG1)	S/U	Yes	Yes
Control 2 Register (CTRL2)	S/U	Yes	No
Error and Status 2 Register (ESR2)	S/U	Yes	Yes
CRC Register (CRCR)	S/U	Yes	Yes
Rx FIFO Global Mask register (RXFGMASK)	S/U	No	No
Rx FIFO Information Register (RXFIR)	S/U	No	No
Message buffers	S/U	No	No
Rx Individual Mask Registers	S/U	No	No
Memory Error Control Register (MECR)	S/U	Yes	Yes
Error Injection Address Register (ERRIAR)	S/U	Yes	Yes
Error Injection Data Pattern Register (ERRIDPR)	S/U	Yes	Yes
Error Injection Parity Pattern Register (ERRIPPR)	S/U	Yes	Yes
Error Report Address Register (RERRAR)	S/U	Yes	Yes
Error Report Data Register (RERRDR)	S/U	Yes	Yes
Error Report Syndrome Register (RERRSYNR)	S/U	Yes	Yes
Error Status Register (ERRSR)	S/U	Yes	Yes

The FlexCAN module can store CAN messages for transmission and reception using mailboxes and Rx FIFO structures.

This module's memory map includes sixty-four 128-bit message buffers (MBs) that occupy the range from offset 0x80 to 0x47F.

CAN memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0000	Module Configuration Register (CAN0_MCR)	32	R/W	5980_000Fh	46.3.2/2438
4002_0004	Control 1 register (CAN0_CTRL1)	32	R/W	0000_0000h	46.3.3/2443
4002_0008	Free Running Timer (CAN0_TIMER)	32	R/W	0000_0000h	46.3.4/2446
4002_0010	Rx Mailboxes Global Mask Register (CAN0_RXMGMASK)	32	R/W	FFFF_FFFFh	46.3.5/2447
4002_0014	Rx 14 Mask register (CAN0_RX14MASK)	32	R/W	FFFF_FFFFh	46.3.6/2448
4002_0018	Rx 15 Mask register (CAN0_RX15MASK)	32	R/W	FFFF_FFFFh	46.3.7/2449
4002_001C	Error Counter (CAN0_ECR)	32	R/W	0000_0000h	46.3.8/2449
4002_0020	Error and Status 1 register (CAN0_ESR1)	32	R/W	0000_0000h	46.3.9/2451
4002_0024	Interrupt Masks 2 register (CAN0_IMASK2)	32	R/W	0000_0000h	46.3.10/ 2455
4002_0028	Interrupt Masks 1 register (CAN0_IMASK1)	32	R/W	0000_0000h	46.3.11/ 2455
4002_002C	Interrupt Flags 2 register (CAN0_IFLAG2)	32	R/W	0000_0000h	46.3.12/ 2456
4002_0030	Interrupt Flags 1 register (CAN0_IFLAG1)	32	R/W	0000_0000h	46.3.13/ 2457
4002_0034	Control 2 register (CAN0_CTRL2)	32	R/W	0080_0000h	46.3.14/ 2459
4002_0038	Error and Status 2 register (CAN0_ESR2)	32	R/W	0000_0000h	46.3.15/ 2462
4002_0044	CRC Register (CAN0_CRCR)	32	R	0000_0000h	46.3.16/ 2464
4002_0048	Rx FIFO Global Mask register (CAN0_RXFGMASK)	32	R/W	FFFF_FFFFh	46.3.17/ 2464
4002_004C	Rx FIFO Information Register (CAN0_RXFIR)	32	R	Undefined	46.3.18/ 2465
4002_0880	Rx Individual Mask Registers (CAN0_RXIMR0)	32	R/W	Undefined	46.3.19/ 2466
4002_0884	Rx Individual Mask Registers (CAN0_RXIMR1)	32	R/W	Undefined	46.3.19/ 2466
4002_0888	Rx Individual Mask Registers (CAN0_RXIMR2)	32	R/W	Undefined	46.3.19/ 2466
4002_088C	Rx Individual Mask Registers (CAN0_RXIMR3)	32	R/W	Undefined	46.3.19/ 2466

Table continues on the next page...

CAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0890	Rx Individual Mask Registers (CAN0_RXIMR4)	32	R/W	Undefined	46.3.19/ 2466
4002_0894	Rx Individual Mask Registers (CAN0_RXIMR5)	32	R/W	Undefined	46.3.19/ 2466
4002_0898	Rx Individual Mask Registers (CAN0_RXIMR6)	32	R/W	Undefined	46.3.19/ 2466
4002_089C	Rx Individual Mask Registers (CAN0_RXIMR7)	32	R/W	Undefined	46.3.19/ 2466
4002_08A0	Rx Individual Mask Registers (CAN0_RXIMR8)	32	R/W	Undefined	46.3.19/ 2466
4002_08A4	Rx Individual Mask Registers (CAN0_RXIMR9)	32	R/W	Undefined	46.3.19/ 2466
4002_08A8	Rx Individual Mask Registers (CAN0_RXIMR10)	32	R/W	Undefined	46.3.19/ 2466
4002_08AC	Rx Individual Mask Registers (CAN0_RXIMR11)	32	R/W	Undefined	46.3.19/ 2466
4002_08B0	Rx Individual Mask Registers (CAN0_RXIMR12)	32	R/W	Undefined	46.3.19/ 2466
4002_08B4	Rx Individual Mask Registers (CAN0_RXIMR13)	32	R/W	Undefined	46.3.19/ 2466
4002_08B8	Rx Individual Mask Registers (CAN0_RXIMR14)	32	R/W	Undefined	46.3.19/ 2466
4002_08BC	Rx Individual Mask Registers (CAN0_RXIMR15)	32	R/W	Undefined	46.3.19/ 2466
4002_08C0	Rx Individual Mask Registers (CAN0_RXIMR16)	32	R/W	Undefined	46.3.19/ 2466
4002_08C4	Rx Individual Mask Registers (CAN0_RXIMR17)	32	R/W	Undefined	46.3.19/ 2466
4002_08C8	Rx Individual Mask Registers (CAN0_RXIMR18)	32	R/W	Undefined	46.3.19/ 2466
4002_08CC	Rx Individual Mask Registers (CAN0_RXIMR19)	32	R/W	Undefined	46.3.19/ 2466
4002_08D0	Rx Individual Mask Registers (CAN0_RXIMR20)	32	R/W	Undefined	46.3.19/ 2466
4002_08D4	Rx Individual Mask Registers (CAN0_RXIMR21)	32	R/W	Undefined	46.3.19/ 2466
4002_08D8	Rx Individual Mask Registers (CAN0_RXIMR22)	32	R/W	Undefined	46.3.19/ 2466
4002_08DC	Rx Individual Mask Registers (CAN0_RXIMR23)	32	R/W	Undefined	46.3.19/ 2466
4002_08E0	Rx Individual Mask Registers (CAN0_RXIMR24)	32	R/W	Undefined	46.3.19/ 2466
4002_08E4	Rx Individual Mask Registers (CAN0_RXIMR25)	32	R/W	Undefined	46.3.19/ 2466

Table continues on the next page...

CAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_08E8	Rx Individual Mask Registers (CAN0_RXIMR26)	32	R/W	Undefined	46.3.19/ 2466
4002_08EC	Rx Individual Mask Registers (CAN0_RXIMR27)	32	R/W	Undefined	46.3.19/ 2466
4002_08F0	Rx Individual Mask Registers (CAN0_RXIMR28)	32	R/W	Undefined	46.3.19/ 2466
4002_08F4	Rx Individual Mask Registers (CAN0_RXIMR29)	32	R/W	Undefined	46.3.19/ 2466
4002_08F8	Rx Individual Mask Registers (CAN0_RXIMR30)	32	R/W	Undefined	46.3.19/ 2466
4002_08FC	Rx Individual Mask Registers (CAN0_RXIMR31)	32	R/W	Undefined	46.3.19/ 2466
4002_0900	Rx Individual Mask Registers (CAN0_RXIMR32)	32	R/W	Undefined	46.3.19/ 2466
4002_0904	Rx Individual Mask Registers (CAN0_RXIMR33)	32	R/W	Undefined	46.3.19/ 2466
4002_0908	Rx Individual Mask Registers (CAN0_RXIMR34)	32	R/W	Undefined	46.3.19/ 2466
4002_090C	Rx Individual Mask Registers (CAN0_RXIMR35)	32	R/W	Undefined	46.3.19/ 2466
4002_0910	Rx Individual Mask Registers (CAN0_RXIMR36)	32	R/W	Undefined	46.3.19/ 2466
4002_0914	Rx Individual Mask Registers (CAN0_RXIMR37)	32	R/W	Undefined	46.3.19/ 2466
4002_0918	Rx Individual Mask Registers (CAN0_RXIMR38)	32	R/W	Undefined	46.3.19/ 2466
4002_091C	Rx Individual Mask Registers (CAN0_RXIMR39)	32	R/W	Undefined	46.3.19/ 2466
4002_0920	Rx Individual Mask Registers (CAN0_RXIMR40)	32	R/W	Undefined	46.3.19/ 2466
4002_0924	Rx Individual Mask Registers (CAN0_RXIMR41)	32	R/W	Undefined	46.3.19/ 2466
4002_0928	Rx Individual Mask Registers (CAN0_RXIMR42)	32	R/W	Undefined	46.3.19/ 2466
4002_092C	Rx Individual Mask Registers (CAN0_RXIMR43)	32	R/W	Undefined	46.3.19/ 2466
4002_0930	Rx Individual Mask Registers (CAN0_RXIMR44)	32	R/W	Undefined	46.3.19/ 2466
4002_0934	Rx Individual Mask Registers (CAN0_RXIMR45)	32	R/W	Undefined	46.3.19/ 2466
4002_0938	Rx Individual Mask Registers (CAN0_RXIMR46)	32	R/W	Undefined	46.3.19/ 2466
4002_093C	Rx Individual Mask Registers (CAN0_RXIMR47)	32	R/W	Undefined	46.3.19/ 2466

Table continues on the next page...

CAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0940	Rx Individual Mask Registers (CAN0_RXIMR48)	32	R/W	Undefined	46.3.19/ 2466
4002_0944	Rx Individual Mask Registers (CAN0_RXIMR49)	32	R/W	Undefined	46.3.19/ 2466
4002_0948	Rx Individual Mask Registers (CAN0_RXIMR50)	32	R/W	Undefined	46.3.19/ 2466
4002_094C	Rx Individual Mask Registers (CAN0_RXIMR51)	32	R/W	Undefined	46.3.19/ 2466
4002_0950	Rx Individual Mask Registers (CAN0_RXIMR52)	32	R/W	Undefined	46.3.19/ 2466
4002_0954	Rx Individual Mask Registers (CAN0_RXIMR53)	32	R/W	Undefined	46.3.19/ 2466
4002_0958	Rx Individual Mask Registers (CAN0_RXIMR54)	32	R/W	Undefined	46.3.19/ 2466
4002_095C	Rx Individual Mask Registers (CAN0_RXIMR55)	32	R/W	Undefined	46.3.19/ 2466
4002_0960	Rx Individual Mask Registers (CAN0_RXIMR56)	32	R/W	Undefined	46.3.19/ 2466
4002_0964	Rx Individual Mask Registers (CAN0_RXIMR57)	32	R/W	Undefined	46.3.19/ 2466
4002_0968	Rx Individual Mask Registers (CAN0_RXIMR58)	32	R/W	Undefined	46.3.19/ 2466
4002_096C	Rx Individual Mask Registers (CAN0_RXIMR59)	32	R/W	Undefined	46.3.19/ 2466
4002_0970	Rx Individual Mask Registers (CAN0_RXIMR60)	32	R/W	Undefined	46.3.19/ 2466
4002_0974	Rx Individual Mask Registers (CAN0_RXIMR61)	32	R/W	Undefined	46.3.19/ 2466
4002_0978	Rx Individual Mask Registers (CAN0_RXIMR62)	32	R/W	Undefined	46.3.19/ 2466
4002_097C	Rx Individual Mask Registers (CAN0_RXIMR63)	32	R/W	Undefined	46.3.19/ 2466
4002_0AE0	Memory Error Control Register (CAN0_MECR)	32	R/W	800C_0080h	46.3.20/ 2467
4002_0AE4	Error Injection Address Register (CAN0_ERRIAR)	32	R/W	0000_0000h	46.3.21/ 2469
4002_0AE8	Error Injection Data Pattern Register (CAN0_ERRIDPR)	32	R/W	0000_0000h	46.3.22/ 2470
4002_0AEC	Error Injection Parity Pattern Register (CAN0_ERRIPPR)	32	R/W	0000_0000h	46.3.23/ 2470
4002_0AF0	Error Report Address Register (CAN0_RERRAR)	32	R	0000_0000h	46.3.24/ 2471
4002_0AF4	Error Report Data Register (CAN0_RERRDR)	32	R/W	0000_0000h	46.3.25/ 2473

Table continues on the next page...

CAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0AF8	Error Report Syndrome Register (CAN0_RERRSYNR)	32	R/W	0000_0000h	46.3.26/ 2473
4002_0AFC	Error Status Register (CAN0_ERRSR)	32	R/W	0000_0000h	46.3.27/ 2476
400D_4000	Module Configuration Register (CAN1_MCR)	32	R/W	5980_000Fh	46.3.2/2438
400D_4004	Control 1 register (CAN1_CTRL1)	32	R/W	0000_0000h	46.3.3/2443
400D_4008	Free Running Timer (CAN1_TIMER)	32	R/W	0000_0000h	46.3.4/2446
400D_4010	Rx Mailboxes Global Mask Register (CAN1_RXMGMASK)	32	R/W	FFFF_FFFFh	46.3.5/2447
400D_4014	Rx 14 Mask register (CAN1_RX14MASK)	32	R/W	FFFF_FFFFh	46.3.6/2448
400D_4018	Rx 15 Mask register (CAN1_RX15MASK)	32	R/W	FFFF_FFFFh	46.3.7/2449
400D_401C	Error Counter (CAN1_ECR)	32	R/W	0000_0000h	46.3.8/2449
400D_4020	Error and Status 1 register (CAN1_ESR1)	32	R/W	0000_0000h	46.3.9/2451
400D_4024	Interrupt Masks 2 register (CAN1_IMASK2)	32	R/W	0000_0000h	46.3.10/ 2455
400D_4028	Interrupt Masks 1 register (CAN1_IMASK1)	32	R/W	0000_0000h	46.3.11/ 2455
400D_402C	Interrupt Flags 2 register (CAN1_IFLAG2)	32	R/W	0000_0000h	46.3.12/ 2456
400D_4030	Interrupt Flags 1 register (CAN1_IFLAG1)	32	R/W	0000_0000h	46.3.13/ 2457
400D_4034	Control 2 register (CAN1_CTRL2)	32	R/W	0080_0000h	46.3.14/ 2459
400D_4038	Error and Status 2 register (CAN1_ESR2)	32	R/W	0000_0000h	46.3.15/ 2462
400D_4044	CRC Register (CAN1_CRCR)	32	R	0000_0000h	46.3.16/ 2464
400D_4048	Rx FIFO Global Mask register (CAN1_RXFGMASK)	32	R/W	FFFF_FFFFh	46.3.17/ 2464
400D_404C	Rx FIFO Information Register (CAN1_RXFIR)	32	R	Undefined	46.3.18/ 2465
400D_4880	Rx Individual Mask Registers (CAN1_RXIMR0)	32	R/W	Undefined	46.3.19/ 2466
400D_4884	Rx Individual Mask Registers (CAN1_RXIMR1)	32	R/W	Undefined	46.3.19/ 2466
400D_4888	Rx Individual Mask Registers (CAN1_RXIMR2)	32	R/W	Undefined	46.3.19/ 2466
400D_488C	Rx Individual Mask Registers (CAN1_RXIMR3)	32	R/W	Undefined	46.3.19/ 2466
400D_4890	Rx Individual Mask Registers (CAN1_RXIMR4)	32	R/W	Undefined	46.3.19/ 2466
400D_4894	Rx Individual Mask Registers (CAN1_RXIMR5)	32	R/W	Undefined	46.3.19/ 2466

Table continues on the next page...

CAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_4898	Rx Individual Mask Registers (CAN1_RXIMR6)	32	R/W	Undefined	46.3.19/ 2466
400D_489C	Rx Individual Mask Registers (CAN1_RXIMR7)	32	R/W	Undefined	46.3.19/ 2466
400D_48A0	Rx Individual Mask Registers (CAN1_RXIMR8)	32	R/W	Undefined	46.3.19/ 2466
400D_48A4	Rx Individual Mask Registers (CAN1_RXIMR9)	32	R/W	Undefined	46.3.19/ 2466
400D_48A8	Rx Individual Mask Registers (CAN1_RXIMR10)	32	R/W	Undefined	46.3.19/ 2466
400D_48AC	Rx Individual Mask Registers (CAN1_RXIMR11)	32	R/W	Undefined	46.3.19/ 2466
400D_48B0	Rx Individual Mask Registers (CAN1_RXIMR12)	32	R/W	Undefined	46.3.19/ 2466
400D_48B4	Rx Individual Mask Registers (CAN1_RXIMR13)	32	R/W	Undefined	46.3.19/ 2466
400D_48B8	Rx Individual Mask Registers (CAN1_RXIMR14)	32	R/W	Undefined	46.3.19/ 2466
400D_48BC	Rx Individual Mask Registers (CAN1_RXIMR15)	32	R/W	Undefined	46.3.19/ 2466
400D_48C0	Rx Individual Mask Registers (CAN1_RXIMR16)	32	R/W	Undefined	46.3.19/ 2466
400D_48C4	Rx Individual Mask Registers (CAN1_RXIMR17)	32	R/W	Undefined	46.3.19/ 2466
400D_48C8	Rx Individual Mask Registers (CAN1_RXIMR18)	32	R/W	Undefined	46.3.19/ 2466
400D_48CC	Rx Individual Mask Registers (CAN1_RXIMR19)	32	R/W	Undefined	46.3.19/ 2466
400D_48D0	Rx Individual Mask Registers (CAN1_RXIMR20)	32	R/W	Undefined	46.3.19/ 2466
400D_48D4	Rx Individual Mask Registers (CAN1_RXIMR21)	32	R/W	Undefined	46.3.19/ 2466
400D_48D8	Rx Individual Mask Registers (CAN1_RXIMR22)	32	R/W	Undefined	46.3.19/ 2466
400D_48DC	Rx Individual Mask Registers (CAN1_RXIMR23)	32	R/W	Undefined	46.3.19/ 2466
400D_48E0	Rx Individual Mask Registers (CAN1_RXIMR24)	32	R/W	Undefined	46.3.19/ 2466
400D_48E4	Rx Individual Mask Registers (CAN1_RXIMR25)	32	R/W	Undefined	46.3.19/ 2466
400D_48E8	Rx Individual Mask Registers (CAN1_RXIMR26)	32	R/W	Undefined	46.3.19/ 2466
400D_48EC	Rx Individual Mask Registers (CAN1_RXIMR27)	32	R/W	Undefined	46.3.19/ 2466

Table continues on the next page...

CAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_48F0	Rx Individual Mask Registers (CAN1_RXIMR28)	32	R/W	Undefined	46.3.19/ 2466
400D_48F4	Rx Individual Mask Registers (CAN1_RXIMR29)	32	R/W	Undefined	46.3.19/ 2466
400D_48F8	Rx Individual Mask Registers (CAN1_RXIMR30)	32	R/W	Undefined	46.3.19/ 2466
400D_48FC	Rx Individual Mask Registers (CAN1_RXIMR31)	32	R/W	Undefined	46.3.19/ 2466
400D_4900	Rx Individual Mask Registers (CAN1_RXIMR32)	32	R/W	Undefined	46.3.19/ 2466
400D_4904	Rx Individual Mask Registers (CAN1_RXIMR33)	32	R/W	Undefined	46.3.19/ 2466
400D_4908	Rx Individual Mask Registers (CAN1_RXIMR34)	32	R/W	Undefined	46.3.19/ 2466
400D_490C	Rx Individual Mask Registers (CAN1_RXIMR35)	32	R/W	Undefined	46.3.19/ 2466
400D_4910	Rx Individual Mask Registers (CAN1_RXIMR36)	32	R/W	Undefined	46.3.19/ 2466
400D_4914	Rx Individual Mask Registers (CAN1_RXIMR37)	32	R/W	Undefined	46.3.19/ 2466
400D_4918	Rx Individual Mask Registers (CAN1_RXIMR38)	32	R/W	Undefined	46.3.19/ 2466
400D_491C	Rx Individual Mask Registers (CAN1_RXIMR39)	32	R/W	Undefined	46.3.19/ 2466
400D_4920	Rx Individual Mask Registers (CAN1_RXIMR40)	32	R/W	Undefined	46.3.19/ 2466
400D_4924	Rx Individual Mask Registers (CAN1_RXIMR41)	32	R/W	Undefined	46.3.19/ 2466
400D_4928	Rx Individual Mask Registers (CAN1_RXIMR42)	32	R/W	Undefined	46.3.19/ 2466
400D_492C	Rx Individual Mask Registers (CAN1_RXIMR43)	32	R/W	Undefined	46.3.19/ 2466
400D_4930	Rx Individual Mask Registers (CAN1_RXIMR44)	32	R/W	Undefined	46.3.19/ 2466
400D_4934	Rx Individual Mask Registers (CAN1_RXIMR45)	32	R/W	Undefined	46.3.19/ 2466
400D_4938	Rx Individual Mask Registers (CAN1_RXIMR46)	32	R/W	Undefined	46.3.19/ 2466
400D_493C	Rx Individual Mask Registers (CAN1_RXIMR47)	32	R/W	Undefined	46.3.19/ 2466
400D_4940	Rx Individual Mask Registers (CAN1_RXIMR48)	32	R/W	Undefined	46.3.19/ 2466
400D_4944	Rx Individual Mask Registers (CAN1_RXIMR49)	32	R/W	Undefined	46.3.19/ 2466

Table continues on the next page...

CAN memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_4948	Rx Individual Mask Registers (CAN1_RXIMR50)	32	R/W	Undefined	46.3.19/ 2466
400D_494C	Rx Individual Mask Registers (CAN1_RXIMR51)	32	R/W	Undefined	46.3.19/ 2466
400D_4950	Rx Individual Mask Registers (CAN1_RXIMR52)	32	R/W	Undefined	46.3.19/ 2466
400D_4954	Rx Individual Mask Registers (CAN1_RXIMR53)	32	R/W	Undefined	46.3.19/ 2466
400D_4958	Rx Individual Mask Registers (CAN1_RXIMR54)	32	R/W	Undefined	46.3.19/ 2466
400D_495C	Rx Individual Mask Registers (CAN1_RXIMR55)	32	R/W	Undefined	46.3.19/ 2466
400D_4960	Rx Individual Mask Registers (CAN1_RXIMR56)	32	R/W	Undefined	46.3.19/ 2466
400D_4964	Rx Individual Mask Registers (CAN1_RXIMR57)	32	R/W	Undefined	46.3.19/ 2466
400D_4968	Rx Individual Mask Registers (CAN1_RXIMR58)	32	R/W	Undefined	46.3.19/ 2466
400D_496C	Rx Individual Mask Registers (CAN1_RXIMR59)	32	R/W	Undefined	46.3.19/ 2466
400D_4970	Rx Individual Mask Registers (CAN1_RXIMR60)	32	R/W	Undefined	46.3.19/ 2466
400D_4974	Rx Individual Mask Registers (CAN1_RXIMR61)	32	R/W	Undefined	46.3.19/ 2466
400D_4978	Rx Individual Mask Registers (CAN1_RXIMR62)	32	R/W	Undefined	46.3.19/ 2466
400D_497C	Rx Individual Mask Registers (CAN1_RXIMR63)	32	R/W	Undefined	46.3.19/ 2466
400D_4AE0	Memory Error Control Register (CAN1_MECR)	32	R/W	800C_0080h	46.3.20/ 2467
400D_4AE4	Error Injection Address Register (CAN1_ERRIAR)	32	R/W	0000_0000h	46.3.21/ 2469
400D_4AE8	Error Injection Data Pattern Register (CAN1_ERRIDPR)	32	R/W	0000_0000h	46.3.22/ 2470
400D_4AEC	Error Injection Parity Pattern Register (CAN1_ERRIPPR)	32	R/W	0000_0000h	46.3.23/ 2470
400D_4AF0	Error Report Address Register (CAN1_RERRAR)	32	R	0000_0000h	46.3.24/ 2471
400D_4AF4	Error Report Data Register (CAN1_RERRDR)	32	R/W	0000_0000h	46.3.25/ 2473
400D_4AF8	Error Report Syndrome Register (CAN1_RERRSYNR)	32	R/W	0000_0000h	46.3.26/ 2473
400D_4AFC	Error Status Register (CAN1_ERRSR)	32	R/W	0000_0000h	46.3.27/ 2476

46.3.2 Module Configuration Register (CANx_MCR)

This register defines global system configurations, such as the module operation modes and the maximum message buffer configuration.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MDIS	FRZ	RFEN	HALT	NOTRDY	0	SOFTRST	FRZACK	SUPV	0	WRNEN	LPMACK	Reserved	0	SRXDIS	IRMQ
W																
Reset	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	LPRIOEN	AEN	0		IDAM		0	MAXMB						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

CANx_MCR field descriptions

Field	Description
31 MDIS	Module Disable This bit controls whether FlexCAN is enabled or not. When disabled, FlexCAN disables the clocks to the CAN Protocol Engine and Controller Host Interface sub-modules. This is the only bit within this register not affected by soft reset. 0 Enable the FlexCAN module. 1 Disable the FlexCAN module.
30 FRZ	Freeze Enable The FRZ bit specifies the FlexCAN behavior when the HALT bit in the MCR Register is set or when Debug mode is requested at MCU level . When FRZ is asserted, FlexCAN is enabled to enter Freeze mode. Negation of this bit field causes FlexCAN to exit from Freeze mode. 0 Not enabled to enter Freeze mode. 1 Enabled to enter Freeze mode.
29 RFEN	Rx FIFO Enable This bit controls whether the Rx FIFO feature is enabled or not. When RFEN is set, MBs 0 to 5 cannot be used for normal reception and transmission because the corresponding memory region (0x80-0xDC) is used by the FIFO engine as well as additional MBs (up to 32, depending on CTRL2[RFFN] setting) which are used as Rx FIFO ID Filter Table elements. RFEN also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in the table "Minimum Ratio Between Peripheral Clock Frequency and CAN Bit Rate" (in section "Arbitration and Matching Timing"). This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Rx FIFO not enabled. 1 Rx FIFO enabled.
28 HALT	Halt FlexCAN

Table continues on the next page...

CANx_MCR field descriptions (continued)

Field	Description
	Assertion of this bit puts the FlexCAN module into Freeze mode. The CPU should clear it after initializing the Message Buffers and Control Register. No reception or transmission is performed by FlexCAN before this bit is cleared. Freeze mode cannot be entered while FlexCAN is in a low power mode. 0 No Freeze mode request. 1 Enters Freeze mode if the FRZ bit is asserted.
27 NOTRDY	FlexCAN Not Ready This read-only bit indicates that FlexCAN is either in Disable mode , Stop mode or Freeze mode. It is negated once FlexCAN has exited these modes. 0 FlexCAN module is either in Normal mode, Listen-Only mode or Loop-Back mode. 1 FlexCAN module is either in Disable mode , Stop mode or Freeze mode.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 SOFTRST	Soft Reset When this bit is asserted, FlexCAN resets its internal state machines and some of the memory mapped registers. The following registers are reset: MCR (except the MDIS bit), TIMER , ECR, ESR1, ESR2, IMASK1, IMASK2, IFLAG1, IFLAG2 and CRCR. Configuration registers that control the interface to the CAN bus are not affected by soft reset. The following registers are unaffected: CTRL1, CTRL2, all RXIMR registers, RXMGMASK, RX14MASK, RX15MASK, RXFGMASK, RXFIR, all Message Buffers . The SOFTRST bit can be asserted directly by the CPU when it writes to the MCR Register, but it is also asserted when global soft reset is requested at MCU level. Because soft reset is synchronous and has to follow a request/acknowledge procedure across clock domains, it may take some time to fully propagate its effect. The SOFTRST bit remains asserted while reset is pending, and is automatically negated when reset completes. Therefore, software can poll this bit to know when the soft reset has completed. Soft reset cannot be applied while clocks are shut down in a low power mode. The module should be first removed from low power mode, and then soft reset can be applied. 0 No reset request. 1 Resets the registers affected by soft reset.
24 FRZACK	Freeze Mode Acknowledge This read-only bit indicates that FlexCAN is in Freeze mode and its prescaler is stopped. The Freeze mode request cannot be granted until current transmission or reception processes have finished. Therefore the software can poll the FRZACK bit to know when FlexCAN has actually entered Freeze mode. If Freeze Mode request is negated, then this bit is negated after the FlexCAN prescaler is running again. If Freeze mode is requested while FlexCAN is in a low power mode, then the FRZACK bit will be set only when the low-power mode is exited. See Section "Freeze Mode". **NOTE:** FRZACK will be asserted within 178 CAN bits from the freeze mode request by the CPU, and negated within 2 CAN bits after the freeze mode request removal (see Section "Protocol Timing"). 0 FlexCAN not in Freeze mode, prescaler running. 1 FlexCAN in Freeze mode, prescaler stopped.
23 SUPV	Supervisor Mode This bit configures the FlexCAN to be either in Supervisor or User mode. The registers affected by this bit are marked as S/U in the Access Type column of the module memory map. Reset value of this bit is 1, so the affected registers start with Supervisor access allowance only . This bit can be written only in Freeze mode because it is blocked by hardware in other modes.

Table continues on the next page...

CANx_MCR field descriptions (continued)

Field	Description
	0 FlexCAN is in User mode. Affected registers allow both Supervisor and Unrestricted accesses . 1 FlexCAN is in Supervisor mode. Affected registers allow only Supervisor access. Unrestricted access behaves as though the access was done to an unimplemented register location .
22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 WRNEN	Warning Interrupt Enable When asserted, this bit enables the generation of the TWRNINT and RWRNINT flags in the Error and Status Register. If WRNEN is negated, the TWRNINT and RWRNINT flags will always be zero, independent of the values of the error counters, and no warning interrupt will ever be generated. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 TWRNINT and RWRNINT bits are zero, independent of the values in the error counters. 1 TWRNINT and RWRNINT bits are set when the respective error counter transitions from less than 96 to greater than or equal to 96.
20 LPMACK	Low-Power Mode Acknowledge This read-only bit indicates that FlexCAN is in a low-power mode (Disable mode , Stop mode). A low-power mode cannot be entered until all current transmission or reception processes have finished, so the CPU can poll the LPMACK bit to know when FlexCAN has actually entered low power mode. **NOTE:** LPMACK will be asserted within 180 CAN bits from the low-power mode request by the CPU, and negated within 2 CAN bits after the low-power mode request removal (see Section "Protocol Timing"). 0 FlexCAN is not in a low-power mode. 1 FlexCAN is in a low-power mode.
19 Reserved	This field is reserved.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 SRXDIS	Self Reception Disable This bit defines whether FlexCAN is allowed to receive frames transmitted by itself. If this bit is asserted, frames transmitted by the module will not be stored in any MB, regardless if the MB is programmed with an ID that matches the transmitted frame, and no interrupt flag or interrupt signal will be generated due to the frame reception. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Self reception enabled. 1 Self reception disabled.
16 IRMQ	Individual Rx Masking And Queue Enable This bit indicates whether Rx matching process will be based either on individual masking and queue or on masking scheme with RXMGMASK, RX14MASK and RX15MASK, RXFGMASK. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Individual Rx masking and queue feature are disabled. For backward compatibility with legacy applications, the reading of C/S word locks the MB even if it is EMPTY. 1 Individual Rx masking and queue feature are enabled.
15 Reserved	This field is reserved.

Table continues on the next page...

CANx_MCR field descriptions (continued)

Field	Description
14 Reserved	This field is reserved.
13 LPRIOEN	Local Priority Enable This bit is provided for backwards compatibility with legacy applications. It controls whether the local priority feature is enabled or not. It is used to expand the ID used during the arbitration process. With this expanded ID concept, the arbitration process is done based on the full 32-bit word, but the actual transmitted ID still has 11-bit for standard frames and 29-bit for extended frames. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Local Priority disabled. 1 Local Priority enabled.
12 AEN	Abort Enable This bit is supplied for backwards compatibility with legacy applications. When asserted, it enables the Tx abort mechanism. This mechanism guarantees a safe procedure for aborting a pending transmission, so that no frame is sent in the CAN bus without notification. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. **NOTE:** When MCR[AEN] is asserted, only the abort mechanism (see Section "Transmission Abort Mechanism") must be used for updating Mailboxes configured for transmission. **CAUTION:** Writing the Abort code into Rx Mailboxes can cause unpredictable results when the MCR[AEN] is asserted. 0 Abort disabled. 1 Abort enabled.
11–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 IDAM	ID Acceptance Mode This 2-bit field identifies the format of the Rx FIFO ID Filter Table elements. Note that all elements of the table are configured at the same time by this field (they are all the same format). See Section "Rx FIFO Structure". This field can be written only in Freeze mode because it is blocked by hardware in other modes. 00 Format A: One full ID (standard and extended) per ID Filter Table element. 01 Format B: Two full standard IDs or two partial 14-bit (standard and extended) IDs per ID Filter Table element. 10 Format C: Four partial 8-bit Standard IDs per ID Filter Table element. 11 Format D: All frames rejected.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 MAXMB	Number Of The Last Message Buffer This 7-bit field defines the number of the last Message Buffers that will take part in the matching and arbitration processes. The reset value (0x0F) is equivalent to a 16 MB configuration. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Number of the last MB = MAXMB **NOTE:** MAXMB must be programmed with a value smaller than the parameter NUMBER_OF_MB, otherwise the number of the last effective Message Buffer will be: (NUMBER_OF_MB - 1)

Table continues on the next page...

CANx_MCR field descriptions (continued)

Field	Description
	Additionally, the value of MAXMB must encompass the FIFO size defined by CTRL2[RFFN]. MAXMB also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table "Minimum Ratio Between Peripheral Clock Frequency and CAN Bit Rate" (in Section "Arbitration and Matching Timing").

46.3.3 Control 1 register (CANx_CTRL1)

This register is defined for specific FlexCAN control features related to the CAN bus, such as bit-rate, programmable sampling point within an Rx bit, Loop Back mode, Listen-Only mode, Bus Off recovery behavior and interrupt enabling (Bus-Off, Error, Warning). It also determines the Division Factor for the clock prescaler.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	PRESDIV								RJW		PSEG1			PSEG2		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BOFFMSK	ERRMSK	CLKSRC	LPB	TWRNMSK	RWRNMSK	0		SMP	BOFFREC	TSYN	LBUF	LOM	PROPSEG		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_CTRL1 field descriptions

Field	Description
31–24 PRESDIV	Prescaler Division Factor This 8-bit field defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency. The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency. The Maximum value of this field is 0xFF, that gives a minimum Sclock frequency equal to the PE clock frequency divided by 256. See Section "Protocol Timing". This field can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (PRESDIV + 1)
23–22 RJW	Resync Jump Width This 2-bit field defines the maximum number of time quanta that a bit time can be changed by one re-synchronization. One time quantum is equal to the Sclock period. The valid programmable values are 0–3. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = RJW + 1.

Table continues on the next page...

CANx_CTRL1 field descriptions (continued)

Field	Description
21–19 PSEG1	Phase Segment 1 This 3-bit field defines the length of Phase Buffer Segment 1 in the bit time. The valid programmable values are 0–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (PSEG1 + 1) × Time-Quanta.
18–16 PSEG2	Phase Segment 2 This 3-bit field defines the length of Phase Buffer Segment 2 in the bit time. The valid programmable values are 1–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 2 = (PSEG2 + 1) × Time-Quanta.
15 BOFFMSK	Bus Off Mask This bit provides a mask for the Bus Off Interrupt. 0 Bus Off interrupt disabled. 1 Bus Off interrupt enabled.
14 ERRMSK	Error Mask This bit provides a mask for the Error Interrupt. 0 Error interrupt disabled. 1 Error interrupt enabled.
13 CLKSRC	CAN Engine Clock Source This bit selects the clock source to the CAN Protocol Engine (PE) to be either the peripheral clock (driven by the PLL) or the crystal oscillator clock. The selected clock is the one fed to the prescaler to generate the Serial Clock (Sclock). In order to guarantee reliable operation, this bit can be written only in Disable mode because it is blocked by hardware in other modes. See Section "Protocol Timing". 0 The CAN engine clock source is the oscillator clock. Under this condition, the oscillator clock frequency must be lower than the bus clock. 1 The CAN engine clock source is the peripheral clock.
12 LPB	Loop Back Mode This bit configures FlexCAN to operate in Loop-Back mode. In this mode, FlexCAN performs an internal loop back that can be used for self test operation. The bit stream output of the transmitter is fed back internally to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic 1). FlexCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field, generating an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. **NOTE:** In this mode, the MCR[SRXDIS] cannot be asserted because this will impede the self reception of a transmitted message. 0 Loop Back disabled. 1 Loop Back enabled.
11 TWRNMSK	Tx Warning Interrupt Mask

Table continues on the next page...

CANx_CTRL1 field descriptions (continued)

Field	Description
	This bit provides a mask for the Tx Warning Interrupt associated with the TWRNINT flag in the Error and Status Register. This bit is read as zero when MCR[WRNEN] bit is negated. This bit can be written only if MCR[WRNEN] bit is asserted. 0 Tx Warning Interrupt disabled. 1 Tx Warning Interrupt enabled.
10 RWRNMSK	Rx Warning Interrupt Mask This bit provides a mask for the Rx Warning Interrupt associated with the RWRNINT flag in the Error and Status Register. This bit is read as zero when MCR[WRNEN] bit is negated. This bit can be written only if MCR[WRNEN] bit is asserted. 0 Rx Warning Interrupt disabled. 1 Rx Warning Interrupt enabled.
9–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 SMP	CAN Bit Sampling This bit defines the sampling mode of CAN bits at the Rx input. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Just one sample is used to determine the bit value. 1 Three samples are used to determine the value of the received bit: the regular one (sample point) and 2 preceding samples; a majority rule is used.
6 BOFFREC	Bus Off Recovery This bit defines how FlexCAN recovers from Bus Off state. If this bit is negated, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If the bit is asserted, automatic recovering from Bus Off is disabled and the module remains in Bus Off state until the bit is negated by the user. If the negation occurs before 128 sequences of 11 recessive bits are detected on the CAN bus, then Bus Off recovery happens as if the BOFFREC bit had never been asserted. If the negation occurs after 128 sequences of 11 recessive bits occurred, then FlexCAN will re-synchronize to the bus by waiting for 11 recessive bits before joining the bus. After negation, the BOFFREC bit can be re-asserted again during Bus Off, but it will be effective only the next time the module enters Bus Off. If BOFFREC was negated when the module entered Bus Off, asserting it during Bus Off will not be effective for the current Bus Off recovery. 0 Automatic recovering from Bus Off state enabled, according to CAN Spec 2.0 part B. 1 Automatic recovering from Bus Off state disabled.
5 TSYN	Timer Sync This bit enables a mechanism that resets the free-running timer each time a message is received in Message Buffer 0. This feature provides means to synchronize multiple FlexCAN stations with a special “SYNC” message, that is, global network time. If the RFEN bit in MCR is set (Rx FIFO enabled), the first available Mailbox, according to CTRL2[RFFN] setting, is used for timer synchronization instead of MB0. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Timer Sync feature disabled 1 Timer Sync feature enabled
4 LBUF	Lowest Buffer Transmitted First

Table continues on the next page...

CANx_CTRL1 field descriptions (continued)

Field	Description
	This bit defines the ordering mechanism for Message Buffer transmission. When asserted, the LPRIOEN bit does not affect the priority arbitration. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Buffer with highest priority is transmitted first. 1 Lowest number buffer is transmitted first.
3 LOM	Listen-Only Mode This bit configures FlexCAN to operate in Listen-Only mode. In this mode, transmission is disabled, all error counters are frozen and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error without changing the REC, as if it was trying to acknowledge the message. Listen-Only mode acknowledgement can be obtained by the state of ESR1[FLTCONF] field which is Passive Error when Listen-Only mode is entered. There can be some delay between the Listen-Only mode request and acknowledge. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Listen-Only mode is deactivated. 1 FlexCAN module operates in Listen-Only mode.
2–0 PROPSEG	Propagation Segment This 3-bit field defines the length of the Propagation Segment in the bit time. The valid programmable values are 0–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation Segment Time = (PROPSEG + 1) × Time-Quanta. Time-Quantum = one Sclock period.

46.3.4 Free Running Timer (CANx_TIMER)

This register represents a 16-bit free running counter that can be read and written by the CPU. The timer starts from 0x0 after Reset, counts linearly to 0xFFFF, and wraps around.

The timer is clocked by the FlexCAN bit-clock, which defines the baud rate on the CAN bus. During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate. The timer is not incremented during Disable , Stop, and Freeze modes.

The timer value is captured when the second bit of the identifier field of any frame is on the CAN bus. This captured value is written into the Time Stamp entry in a message buffer after a successful reception or transmission of a message.

If bit CTRL1[TSYN] is asserted, the Timer is reset whenever a message is received in the first available Mailbox, according to CTRL2[RFFN] setting.

The CPU can write to this register anytime. However, if the write occurs at the same time that the Timer is being reset by a reception in the first Mailbox, then the write value is discarded.

Reading this register affects the Mailbox Unlocking procedure; see Section "Mailbox Lock Mechanism".

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TIMER															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_TIMER field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 TIMER	Timer Value Contains the free-running counter value.

46.3.5 Rx Mailboxes Global Mask Register (CANx_RXMGMASK)

This register is located in RAM.

RXMGMASK is provided for legacy application support.

- When the MCR[IRMQ] bit is negated, RXMGMASK is always in effect.
- When the MCR[IRMQ] bit is asserted, RXMGMASK has no effect.

RXMGMASK is used to mask the filter fields of all Rx MBs, excluding MBs 14-15, which have individual mask registers.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MG[31:0]																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CANx_RXMGMASK field descriptions

Field	Description
31–0 MG[31:0]	Rx Mailboxes Global Mask Bits

CANx_RXMGMASK field descriptions (continued)

Field	Description
	These bits mask the Mailbox filter bits. Note that the alignment with the ID word of the Mailbox is not perfect as the two most significant MG bits affect the fields RTR and IDE, which are located in the Control and Status word of the Mailbox. The following table shows in detail which MG bits mask each Mailbox filter field.

SMB[RTR] [1]	CTRL2[RRS]	CTRL2[EACEN]	Mailbox filter fields			
			MB[RTR]	MB[IDE]	MB[ID]	Reserved
0	-	0	note [2]	note [3]	MG[28:0]	MG[31:29]
0	-	1	MG[31]	MG[30]	MG[28:0]	MG[29]
1	0	-	-	-	-	MG[31:0]
1	1	0	-	-	MG[28:0]	MG[31:29]
1	1	1	MG[31]	MG[30]	MG[28:0]	MG[29]

1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If the CTRL2[EACEN] bit is negated, the RTR bit of Mailbox is never compared with the RTR bit of the incoming frame.
3. If the CTRL2[EACEN] bit is negated, the IDE bit of Mailbox is always compared with the IDE bit of the incoming frame.

0 The corresponding bit in the filter is "don't care."
1 The corresponding bit in the filter is checked.

1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If the CTRL2[EACEN] bit is negated, the RTR bit of Mailbox is never compared with the RTR bit of the incoming frame.
3. If the CTRL2[EACEN] bit is negated, the IDE bit of Mailbox is always compared with the IDE bit of the incoming frame.

46.3.6 Rx 14 Mask register (CANx_RX14MASK)

This register is located in RAM.

RX14MASK is provided for legacy application support. When the MCR[IRMQ] bit is asserted, RX14MASK has no effect.

RX14MASK is used to mask the filter fields of Message Buffer 14.

This register can only be programmed while the module is in Freeze mode as it is blocked by hardware in other modes.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX14M[31:0]																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CANx_RX14MASK field descriptions

Field	Description
31–0 RX14M[31:0]	Rx Buffer 14 Mask Bits Each mask bit masks the corresponding Mailbox 14 filter field in the same way that RXMGMASK masks other Mailboxes' filters. See the description of the CAN_RXMGMASK register. 0 The corresponding bit in the filter is "don’t care." 1 The corresponding bit in the filter is checked.

46.3.7 Rx 15 Mask register (CANx_RX15MASK)

This register is located in RAM.

RX15MASK is provided for legacy application support. When the MCR[IRMQ] bit is asserted, RX15MASK has no effect.

RX15MASK is used to mask the filter fields of Message Buffer 15.

This register can be programmed only while the module is in Freeze mode because it is blocked by hardware in other modes.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX15M[31:0]																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CANx_RX15MASK field descriptions

Field	Description
31–0 RX15M[31:0]	Rx Buffer 15 Mask Bits Each mask bit masks the corresponding Mailbox 15 filter field in the same way that RXMGMASK masks other Mailboxes' filters. See the description of the CAN_RXMGMASK register. 0 The corresponding bit in the filter is "don’t care." 1 The corresponding bit in the filter is checked.

46.3.8 Error Counter (CANx_ECR)

This register has two 8-bit fields reflecting the value of two FlexCAN error counters: Transmit Error Counter (TXERRCNT field) and Receive Error Counter (RXERRCNT field). The rules for increasing and decreasing these counters are described in the CAN protocol and are completely implemented in the FlexCAN module. Both counters are read-only except in Freeze mode, where they can be written by the CPU.

FlexCAN responds to any bus state as described in the protocol, for example, transmit Error Active or Error Passive flag, delay its transmission start time (Error Passive) and avoid any influence on the bus when in Bus Off state.

The following are the basic rules for FlexCAN bus state transitions:

- If the value of TXERRCNT or RXERRCNT increases to be greater than or equal to 128, the FLTCONF field in the Error and Status Register is updated to reflect ‘Error Passive’ state.
- If the FlexCAN state is ‘Error Passive’, and either TXERRCNT or RXERRCNT decrements to a value less than or equal to 127 while the other already satisfies this condition, the FLTCONF field in the Error and Status Register is updated to reflect ‘Error Active’ state.
- If the value of TXERRCNT increases to be greater than 255, the FLTCONF field in the Error and Status Register is updated to reflect ‘Bus Off’ state, and an interrupt may be issued. The value of TXERRCNT is then reset to zero.
- If FlexCAN is in ‘Bus Off’ state, then TXERRCNT is cascaded together with another internal counter to count the 128th occurrences of 11 consecutive recessive bits on the bus. Hence, TXERRCNT is reset to zero and counts in a manner where the internal counter counts 11 such bits and then wraps around while incrementing the TXERRCNT. When TXERRCNT reaches the value of 128, the FLTCONF field in the Error and Status Register is updated to be ‘Error Active’ and both error counters are reset to zero. At any instance of dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the TXERRCNT value.
- If during system start-up, only one node is operating, then its TXERRCNT increases in each message it is trying to transmit, as a result of acknowledge errors (indicated by the ACKERR bit in the Error and Status Register). After the transition to ‘Error Passive’ state, the TXERRCNT does not increment anymore by acknowledge errors. Therefore the device never goes to the ‘Bus Off’ state.
- If the RXERRCNT increases to a value greater than 127, it is not incremented further, even if more errors are detected while being a receiver. At the next successful message reception, the counter is set to a value between 119 and 127 to resume to ‘Error Active’ state.

Address: Base address + 1Ch offset

Bit	31–16	15–8	7–0
R	0	RXERRCNT	TXERRCNT
W			
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

CANx_ECR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 RXERRCNT	Receive Error Counter
7–0 TXERRCNT	Transmit Error Counter

46.3.9 Error and Status 1 register (CANx_ESR1)

This register reflects various error conditions, some general status of the device and it is the source of interrupts to the CPU.

The CPU read action clears bits 15-10. Therefore the reported error conditions (bits 15-10) are those that occurred since the last time the CPU read this register. Bits 9-3 are status bits.

The following table shows the FlexCAN state variables and their meanings. Other combinations not shown in the table are reserved.

SYNCH	IDLE	TX	RX	FlexCAN State
0	0	0	0	Not synchronized to CAN bus
1	1	x	x	Idle
1	0	1	0	Transmitting
1	0	0	1	Receiving

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													SYNCH	TWRNINT	RWRNINT
W															w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BIT1ERR	BIT0ERR	ACKERR	CRCERR	FRMERR	STFERR	TXWRN	RXWRN	IDLE	TX	FLTCONF		RX	BOFFINT	ERRINT	0
W														w1c	w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_ESR1 field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 SYNCH	CAN Synchronization Status This read-only flag indicates whether the FlexCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the FlexCAN. See the table in the overall CAN_ESR1 register description. 0 FlexCAN is not synchronized to the CAN bus. 1 FlexCAN is synchronized to the CAN bus.
17 TWRNINT	Tx Warning Interrupt Flag If the WRNEN bit in MCR is asserted, the TWRNINT bit is set when the TXWRN flag transitions from 0 to 1, meaning that the Tx error counter reached 96. If the corresponding mask bit in the Control Register (TWRNMSK) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When WRNEN is negated, this flag is masked. CPU must clear this flag before disabling the bit. Otherwise it will be set when the WRNEN is set again. Writing 0 has no effect. This flag is not generated during Bus Off state. This bit is not updated during Freeze mode. 0 No such occurrence. 1 The Tx error counter transitioned from less than 96 to greater than or equal to 96.
16 RWRNINT	Rx Warning Interrupt Flag If the WRNEN bit in MCR is asserted, the RWRNINT bit is set when the RXWRN flag transitions from 0 to 1, meaning that the Rx error counters reached 96. If the corresponding mask bit in the Control Register (RWRNMSK) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When WRNEN is negated, this flag is masked. CPU must clear this flag before disabling the bit. Otherwise it will be set when the WRNEN is set again. Writing 0 has no effect. This bit is not updated during Freeze mode. 0 No such occurrence. 1 The Rx error counter transitioned from less than 96 to greater than or equal to 96.
15 BIT1ERR	Bit1 Error This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message.

Table continues on the next page...

CANx_ESR1 field descriptions (continued)

Field	Description
	NOTE: This bit is not set by a transmitter in case of arbitration field or ACK slot, or in case of a node sending a passive error flag that detects dominant bits. 0 No such occurrence. 1 At least one bit sent as recessive is received as dominant.
14 BIT0ERR	Bit0 Error This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message. 0 No such occurrence. 1 At least one bit sent as dominant is received as recessive.
13 ACKERR	Acknowledge Error This bit indicates that an Acknowledge Error has been detected by the transmitter node, that is, a dominant bit has not been detected during the ACK SLOT. 0 No such occurrence. 1 An ACK error occurred since last read of this register.
12 CRCERR	Cyclic Redundancy Check Error This bit indicates that a CRC Error has been detected by the receiver node, that is, the calculated CRC is different from the received. 0 No such occurrence. 1 A CRC error occurred since last read of this register.
11 FRMERR	Form Error This bit indicates that a Form Error has been detected by the receiver node, that is, a fixed-form bit field contains at least one illegal bit. 0 No such occurrence. 1 A Form Error occurred since last read of this register.
10 STFERR	Stuffing Error This bit indicates that a Stuffing Error has been etected. 0 No such occurrence. 1 A Stuffing Error occurred since last read of this register.
9 TXWRN	TX Error Warning This bit indicates when repetitive errors are occurring during message transmission. This bit is not updated during Freeze mode. 0 No such occurrence. 1 TXERRCNT is greater than or equal to 96.
8 RXWRN	Rx Error Warning This bit indicates when repetitive errors are occurring during message reception. This bit is not updated during Freeze mode. 0 No such occurrence. 1 RXERRCNT is greater than or equal to 96.

Table continues on the next page...

CANx_ESR1 field descriptions (continued)

Field	Description
7 IDLE	This bit indicates when CAN bus is in IDLE state. See the table in the overall CAN_ESR1 register description. 0 No such occurrence. 1 CAN bus is now IDLE.
6 TX	FlexCAN In Transmission This bit indicates if FlexCAN is transmitting a message. See the table in the overall CAN_ESR1 register description. 0 FlexCAN is not transmitting a message. 1 FlexCAN is transmitting a message.
5–4 FLTCONF	Fault Confinement State This 2-bit field indicates the Confinement State of the FlexCAN module. If the LOM bit in the Control Register is asserted, after some delay that depends on the CAN bit timing the FLTCONF field will indicate “Error Passive”. The very same delay affects the way how FLTCONF reflects an update to ECR register by the CPU. It may be necessary up to one CAN bit time to get them coherent again. Because the Control Register is not affected by soft reset, the FLTCONF field will not be affected by soft reset if the LOM bit is asserted. 00 Error Active 01 Error Passive 1x Bus Off
3 RX	FlexCAN In Reception This bit indicates if FlexCAN is receiving a message. See the table in the overall CAN_ESR1 register description. 0 FlexCAN is not receiving a message. 1 FlexCAN is receiving a message.
2 BOFFINT	Bus Off Interrupt This bit is set when FlexCAN enters ‘Bus Off’ state. If the corresponding mask bit in the Control Register (BOFFMSK) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0 No such occurrence. 1 FlexCAN module entered Bus Off state.
1 ERRINT	Error Interrupt This bit indicates that at least one of the Error Bits (bits 15-10) is set. If the corresponding mask bit CTRL1[ERRMSK] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect. 0 No such occurrence. 1 Indicates setting of any Error Bit in the Error and Status Register.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

46.3.10 Interrupt Masks 2 register (CANx_IMASK2)

This register allows any number of a range of the 32 Message Buffer Interrupts to be enabled or disabled for MB63 to MB32. It contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception, that is, when the corresponding IFLAG2 bit is set.

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUFHM																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_IMASK2 field descriptions

Field	Description
31–0 BUFHM	Buffer MB $_i$ Mask Each bit enables or disables the corresponding FlexCAN Message Buffer Interrupt for MB63 to MB32. **NOTE:** Setting or clearing a bit in the IMASK2 Register can assert or negate an interrupt request, if the corresponding IFLAG2 bit is set. 0 The corresponding buffer Interrupt is disabled. 1 The corresponding buffer Interrupt is enabled.

46.3.11 Interrupt Masks 1 register (CANx_IMASK1)

This register allows any number of a range of the 32 Message Buffer Interrupts to be enabled or disabled for MB31 to MB0. It contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception, that is, when the corresponding IFLAG1 bit is set.

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUFLM																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_IMASK1 field descriptions

Field	Description
31–0 BUFLM	Buffer MB $_i$ Mask Each bit enables or disables the corresponding FlexCAN Message Buffer Interrupt for MB31 to MB0. **NOTE:** Setting or clearing a bit in the IMASK1 Register can assert or negate an interrupt request, if the corresponding IFLAG1 bit is set. 0 The corresponding buffer Interrupt is disabled. 1 The corresponding buffer Interrupt is enabled.

46.3.12 Interrupt Flags 2 register (CANx_IFLAG2)

This register defines the flags for the 32 Message Buffer interrupts for MB63 to MB32. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG2 bit. If the corresponding IMASK2 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.

Before updating MCR[MAXMB] field, CPU must service the IFLAG2 bits whose MB value is greater than the MCR[MAXMB] to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

Address: Base address + 2Ch offset

Bit	31–0
R	BUFHI
W	w1c
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

CANx_IFLAG2 field descriptions

Field	Description
31–0 BUFHI	Buffer MB $_i$ Interrupt Each bit flags the corresponding FlexCAN Message Buffer interrupt for MB63 to MB32. 0 The corresponding buffer has no occurrence of successfully completed transmission or reception. 1 The corresponding buffer has successfully completed transmission or reception.

46.3.13 Interrupt Flags 1 register (CANx_IFLAG1)

This register defines the flags for the 32 Message Buffer interrupts for MB31 to MB0. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG1 bit. If the corresponding IMASK1 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.

The BUF7I to BUF5I flags are also used to represent FIFO interrupts when the Rx FIFO is enabled. When the bit MCR[RFEN] is set, the function of the 8 least significant interrupt flags BUF[7:0]I changes: BUF7I, BUF6I and BUF5I indicate operating conditions of the FIFO, and the BUF4TO0I field is reserved.

Before enabling the RFEN, the CPU must service the IFLAG bits asserted in the Rx FIFO region; see Section "Rx FIFO". Otherwise, these IFLAG bits will mistakenly show the related MBs now belonging to FIFO as having contents to be serviced. When the RFEN bit is negated, the FIFO flags must be cleared. The same care must be taken when an RFFN value is selected extending Rx FIFO filters beyond MB7. For example, when RFFN is 0x8, the MB0-23 range is occupied by Rx FIFO filters and related IFLAG bits must be cleared.

Before updating MCR[MAXMB] field, CPU must service the IFLAG1 bits whose MB value is greater than the MCR[MAXMB] to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF31TO8I															
W	w1c															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF31TO8I								BUF7I	BUF6I	BUF5I	BUF4TO1I				BUF0I
W	w1c								w1c	w1c	w1c	w1c				w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_IFLAG1 field descriptions

Field	Description
31–8 BUF31TO8I	Buffer MB_i Interrupt Each bit flags the corresponding FlexCAN Message Buffer interrupt for MB31 to MB8. 0 The corresponding buffer has no occurrence of successfully completed transmission or reception. 1 The corresponding buffer has successfully completed transmission or reception.
7 BUF7I	Buffer MB7 Interrupt Or "Rx FIFO Overflow" When the RFEN bit in the MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB7. **NOTE:** This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes. The BUF7I flag represents "Rx FIFO Overflow" when MCR[RFEN] is set. In this case, the flag indicates that a message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox. 0 No occurrence of MB7 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO overflow when MCR[RFEN]=1 1 MB7 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO overflow when MCR[RFEN]=1
6 BUF6I	Buffer MB6 Interrupt Or "Rx FIFO Warning" When the RFEN bit in the MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB6. **NOTE:** This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes. The BUF6I flag represents "Rx FIFO Warning" when MCR[RFEN] is set. In this case, the flag indicates when the number of unread messages within the Rx FIFO is increased to 5 from 4 due to the reception of a new one, meaning that the Rx FIFO is almost full. Note that if the flag is cleared while the number of unread messages is greater than 4, it does not assert again until the number of unread messages within the Rx FIFO is decreased to be equal to or less than 4. 0 No occurrence of MB6 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO almost full when MCR[RFEN]=1 1 MB6 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO almost full when MCR[RFEN]=1
5 BUF5I	Buffer MB5 Interrupt Or "Frames available in Rx FIFO" When the RFEN bit in the MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB5. **NOTE:** This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes. The BUF5I flag represents "Frames available in Rx FIFO" when MCR[RFEN] is set. In this case, the flag indicates that at least one frame is available to be read from the Rx FIFO. 0 No occurrence of MB5 completing transmission/reception when MCR[RFEN]=0, or of frame(s) available in the FIFO, when MCR[RFEN]=1 1 MB5 completed transmission/reception when MCR[RFEN]=0, or frame(s) available in the Rx FIFO when MCR[RFEN]=1
4–1 BUF4TO1I	Buffer MB_i Interrupt Or "reserved" When the RFEN bit in the MCR is cleared (Rx FIFO disabled), these bits flag the interrupts for MB4 to MB1. **NOTE:** These flags are cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes. The BUF4TO1I flags are reserved when MCR[RFEN] is set.

Table continues on the next page...

CANx_IFLAG1 field descriptions (continued)

Field	Description
	0 The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0. 1 The corresponding buffer has successfully completed transmission or reception when MCR[RFEN]=0.
0 BUF0I	Buffer MB0 Interrupt Or "reserved" When the RFEN bit in the MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB0. **NOTE:** This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes. The BUF0I flag is reserved when MCR[RFEN] is set. 0 The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0. 1 The corresponding buffer has successfully completed transmission or reception when MCR[RFEN]=0.

46.3.14 Control 2 register (CANx_CTRL2)

This register contains control bits for CAN errors, FIFO features, and mode selection.

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		ECRWRE	WRMFRZ	RFFN				TASD					MRP	RRS	EACEN
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_CTRL2 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 ECRWRE	Error-correction Configuration Register Write Enable Enable that the MECR register is updated. This bit is automatically set to 0 if the protocol described in the "Detection and Correction of Memory Errors" section is not followed. 0 Disable update. 1 Enable update.
28 WRMFRZ	Write-Access To Memory In Freeze Mode

Table continues on the next page...

CANx_CTRL2 field descriptions (continued)

Field	Description
	Enable unrestricted write access to FlexCAN memory in Freeze mode. This bit can only be written in Freeze mode and has no effect out of Freeze mode. 0 Maintain the write access restrictions. 1 Enable unrestricted write access to FlexCAN memory.
27–24 RFFN	Number Of Rx FIFO Filters This 4-bit field defines the number of Rx FIFO filters, as shown in the following table. The maximum selectable number of filters is determined by the MCU. This field can only be written in Freeze mode as it is blocked by hardware in other modes. This field must not be programmed with values that make the number of Message Buffers occupied by Rx FIFO and ID Filter exceed the number of Mailboxes present, defined by MCR[MAXMB]. **NOTE:** Each group of eight filters occupies a memory space equivalent to two Message Buffers which means that the more filters are implemented the less Mailboxes will be available. Considering that the Rx FIFO occupies the memory space originally reserved for MB0-5, RFFN should be programmed with a value correponding to a number of filters not greater than the number of available memory words which can be calculated as follows: (SETUP_MB - 6) × 4 where SETUP_MB is the least between NUMBER_OF_MB and MAXMB. The number of remaining Mailboxes available will be: (SETUP_MB - 8) - (RFFN × 2) If the Number of Rx FIFO Filters programmed through RFFN exceeds the SETUP_MB value (memory space available) the exceeding ones will not be functional.

RFFN[3:0]	Number of Rx FIFO filters	Message Buffers occupied by Rx FIFO and ID Filter Table	Remaining Available Mailboxes[1]	Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks[2]	Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask [2]
0x0	8	MB 0-7	MB 8-63	Elements 0-7	none
0x1	16	MB 0-9	MB 10-63	Elements 0-9	Elements 10-15
0x2	24	MB 0-11	MB 12-63	Elements 0-11	Elements 12-23
0x3	32	MB 0-13	MB 14-63	Elements 0-13	Elements 14-31
0x4	40	MB 0-15	MB 16-63	Elements 0-15	Elements 16-39
0x5	48	MB 0-17	MB 18-63	Elements 0-17	Elements 18-47
0x6	56	MB 0-19	MB 20-63	Elements 0-19	Elements 20-55
0x7	64	MB 0-21	MB 22-63	Elements 0-21	Elements 22-63
0x8	72	MB 0-23	MB 24-63	Elements 0-23	Elements 24-71
0x9	80	MB 0-25	MB 26-63	Elements 0-25	Elements 26-79
0xA	88	MB 0-27	MB 28-63	Elements 0-27	Elements 28-87
0xB	96	MB 0-29	MB 30-63	Elements 0-29	Elements 30-95
0xC	104	MB 0-31	MB 32-63	Elements 0-31	Elements 32-103
0xD	112	MB 0-33	MB 34-63	Elements 0-31	Elements 32-111
0xE	120	MB 0-35	MB 36-63	Elements 0-31	Elements 32-119

Table continues on the next page...

CANx_CTRL2 field descriptions (continued)

Field	Description
	(continued from previous page)

RFFN[3:0]	Number of Rx FIFO filters	Message Buffers occupied by Rx FIFO and ID Filter Table	Remaining Available Mailboxes[1]	Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks[2]	Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask [2]
0xF	128	MB 0-37	MB 38-63	Elements 0-31	Elements 32-127

1. The number of the last remaining available mailboxes is defined by the least value between the parameter NUMBER_OF_MB minus 1 and the MCR[MAXMB] field.
2. If Rx Individual Mask Registers are not enabled then all Rx FIFO filters are affected by the Rx FIFO Global Mask.

23–19 TASD

Tx Arbitration Start Delay

This 5-bit field indicates how many CAN bits the Tx arbitration process start point can be delayed from the first bit of CRC field on CAN bus. This field can be written only in Freeze mode because it is blocked by hardware in other modes.

This field is useful to optimize the transmit performance based on factors such as: peripheral/serial clock ratio, CAN bit timing and number of MBs. The duration of an arbitration process, in terms of CAN bits, is directly proportional to the number of available MBs and CAN baud rate and inversely proportional to the peripheral clock frequency.

The optimal arbitration timing is that in which the last MB is scanned right before the first bit of the Intermission field of a CAN frame. Therefore, if there are few MBs and the system/serial clock ratio is high and the CAN baud rate is low then the arbitration can be delayed and vice-versa.

If TASD is 0 then the arbitration start is not delayed, thus the CPU has less time to configure a Tx MB for the next arbitration, but more time is reserved for arbitration. On the other hand, if TASD is 24 then the CPU can configure a Tx MB later and less time is reserved for arbitration.

If too little time is reserved for arbitration the FlexCAN may be not able to find winner MBs in time to compete with other nodes for the CAN bus. If the arbitration ends too much time before the first bit of Intermission field then there is a chance that the CPU reconfigures some Tx MBs and the winner MB is not the best to be transmitted.

The optimal configuration for TASD can be calculated as:

$$TASD = 25 - \{f_{CANCLK} \times [MAXMB + 3 - (RFEN \times 8) - (RFEN \times RFFN \times 2)] \times 2\} / \{f_{SYS} \times [1+(PSEG1+1)+(PSEG2+1)+(PROPSEG+1)] \times (PRESDIV+1)\}$$

where:

- f_{CANCLK} is the Protocol Engine (PE) Clock (see section "Protocol Timing"), in Hz
- f_{SYS} is the peripheral clock, in Hz
- MAXMB is the value in CTRL1[MAXMB] field
- RFEN is the value in CTRL1[RFEN] bit
- RFFN is the value in CTRL2[RFFN] field
- PSEG1 is the value in CTRL1[PSEG1] field
- PSEG2 is the value in CTRL1[PSEG2] field
- PROPSEG is the value in CTRL1[PROPSEG] field
- PRESDIV is the value in CTRL1[PRESDIV] field

See Section "Arbitration process" and Section "Protocol Timing" for more details.

Table continues on the next page...

CANx_CTRL2 field descriptions (continued)

Field	Description
18 MRP	Mailboxes Reception Priority If this bit is set the matching process starts from the Mailboxes and if no match occurs the matching continues on the Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Matching starts from Rx FIFO and continues on Mailboxes. 1 Matching starts from Mailboxes and continues on Rx FIFO.
17 RRS	Remote Request Storing If this bit is asserted Remote Request Frame is submitted to a matching process and stored in the corresponding Message Buffer in the same fashion of a Data Frame. No automatic Remote Response Frame will be generated. If this bit is negated the Remote Request Frame is submitted to a matching process and an automatic Remote Response Frame is generated if a Message Buffer with CODE=0b1010 is found with the same ID. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Remote Response Frame is generated. 1 Remote Request Frame is stored.
16 EACEN	Entire Frame Arbitration Field Comparison Enable For Rx Mailboxes This bit controls the comparison of IDE and RTR bits whithin Rx Mailboxes filters with their corresponding bits in the incoming frame by the matching process. This bit does not affect matching for Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes. 0 Rx Mailbox filter's IDE bit is always compared and RTR is never compared despite mask bits. 1 Enables the comparison of both Rx Mailbox filter's IDE and RTR bit with their corresponding bits within the incoming frame. Mask bits do apply.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

1. The number of the last remaining available mailboxes is defined by the least value between the parameter NUMBER_OF_MB minus 1 and the MCR[MAXMB] field.
2. If Rx Individual Mask Registers are not enabled then all Rx FIFO filters are affected by the Rx FIFO Global Mask.

46.3.15 Error and Status 2 register (CANx_ESR2)

This register reflects various interrupt flags and some general status.

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									LPTM						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	VPS	IMB	0												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_ESR2 field descriptions

Field	Description
31–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–16 LPTM	Lowest Priority Tx Mailbox If ESR2[VPS] is asserted, this field indicates the lowest number inactive Mailbox (see the IMB bit description). If there is no inactive Mailbox then the Mailbox indicated depends on CTRL1[LBUF] bit value. If CTRL1[LBUF] bit is negated then the Mailbox indicated is the one that has the greatest arbitration value (see the "Highest priority Mailbox first" section). If CTRL1[LBUF] bit is asserted then the Mailbox indicated is the highest number active Tx Mailbox. If a Tx Mailbox is being transmitted it is not considered in LPTM calculation. If ESR2[IMB] is not asserted and a frame is transmitted successfully, LPTM is updated with its Mailbox number.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 VPS	Valid Priority Status This bit indicates whether IMB and LPTM contents are currently valid or not. VPS is asserted upon every complete Tx arbitration process unless the CPU writes to Control and Status word of a Mailbox that has already been scanned, that is, it is behind Tx Arbitration Pointer, during the Tx arbitration process. If there is no inactive Mailbox and only one Tx Mailbox that is being transmitted then VPS is not asserted. VPS is negated upon the start of every Tx arbitration process or upon a write to Control and Status word of any Mailbox. **NOTE:** ESR2[VPS] is not affected by any CPU write into Control Status (C/S) of a MB that is blocked by abort mechanism. When MCR[AEN] is asserted, the abort code write in C/S of a MB that is being transmitted (pending abort), or any write attempt into a Tx MB with IFLAG set is blocked. 0 Contents of IMB and LPTM are invalid. 1 Contents of IMB and LPTM are valid.
13 IMB	Inactive Mailbox If ESR2[VPS] is asserted, this bit indicates whether there is any inactive Mailbox (CODE field is either 0b1000 or 0b0000). This bit is asserted in the following cases: • During arbitration, if an LPTM is found and it is inactive. • If IMB is not asserted and a frame is transmitted successfully. This bit is cleared in all start of arbitration (see Section "Arbitration process"). **NOTE:** LPTM mechanism have the following behavior: if an MB is successfully transmitted and ESR2[IMB]=0 (no inactive Mailbox), then ESR2[VPS] and ESR2[IMB] are asserted and the index related to the MB just transmitted is loaded into ESR2[LPTM]. 0 If ESR2[VPS] is asserted, the ESR2[LPTM] is not an inactive Mailbox. 1 If ESR2[VPS] is asserted, there is at least one inactive Mailbox. LPTM content is the number of the first one.
12–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

46.3.16 CRC Register (CANx_CRCR)

This register provides information about the CRC of transmitted messages.

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									MBCRC						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TXCRC														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_CRCR field descriptions

Field	Description
31–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–16 MBCRC	CRC Mailbox This field indicates the number of the Mailbox corresponding to the value in TXCRC field.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–0 TXCRC	CRC Transmitted This field indicates the CRC value of the last message transmitted. This field is updated at the same time the Tx Interrupt Flag is asserted.

46.3.17 Rx FIFO Global Mask register (CANx_RXFGMASK)

This register is located in RAM.

If Rx FIFO is enabled RXFGMASK is used to mask the Rx FIFO ID Filter Table elements that do not have a corresponding RXIMR according to CTRL2[RFFN] field setting.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

Address: Base address + 48h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FGM[31:0]																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CANx_RXFGMASK field descriptions

Field	Description
31–0 FGM[31:0]	Rx FIFO Global Mask Bits These bits mask the ID Filter Table elements bits in a perfect alignment. The following table shows how the FGM bits correspond to each IDAF field.

Rx FIFO ID Filter Table Elements Format (MCR[IDAM])	Identifier Acceptance Filter Fields					
	RTR	IDE	RXIDA	RXIDB [1]	RXIDC [2]	Reserved
A	FGM[31]	FGM[30]	FGM[29:1]	-	-	FGM[0]
B	FGM[31], FGM[15]	FGM[30], FGM[14]	-	FGM[29:16], FGM[13:0]		-
C	-	-		-	FGM[31:24], FGM[23:16], FGM[15:8], FGM[7:0]	

1. If MCR[IDAM] field is equivalent to the format B only the fourteen most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.
2. If MCR[IDAM] field is equivalent to the format C only the eight most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.

0 The corresponding bit in the filter is "don't care."
1 The corresponding bit in the filter is checked.

1. If MCR[IDAM] field is equivalent to the format B only the fourteen most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.
2. If MCR[IDAM] field is equivalent to the format C only the eight most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.

46.3.18 Rx FIFO Information Register (CANx_RXFIR)

RXFIR provides information on Rx FIFO.

This register is the port through which the CPU accesses the output of the RXFIR FIFO located in RAM. The RXFIR FIFO is written by the FlexCAN whenever a new message is moved into the Rx FIFO as well as its output is updated whenever the output of the Rx FIFO is updated with the next message. See Section "Rx FIFO" for instructions on reading this register.

Address: Base address + 4Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																							IDHIT								
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CANx_RXFIR field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–0 IDHIT	Identifier Acceptance Filter Hit Indicator This field indicates which Identifier Acceptance Filter was hit by the received message that is in the output of the Rx FIFO. If multiple filters match the incoming message ID then the first matching IDAF found (lowest number) by the matching process is indicated. This field is valid only while the IFLAG[BUF5I] is asserted.

46.3.19 Rx Individual Mask Registers (CANx_RXIMR*n*)

These registers are located in RAM.

RXIMR are used as acceptance masks for ID filtering in Rx MBs and the Rx FIFO. If the Rx FIFO is not enabled, one mask register is provided for each available Mailbox, providing ID masking capability on a per Mailbox basis.

When the Rx FIFO is enabled (MCR[RFEN] bit is asserted), up to 32 Rx Individual Mask Registers can apply to the Rx FIFO ID Filter Table elements on a one-to-one correspondence depending on the setting of CTRL2[RFFN].

RXIMR can only be written by the CPU while the module is in Freeze mode; otherwise, they are blocked by hardware.

The Individual Rx Mask Registers are not affected by reset and must be explicitly initialized prior to any reception.

Address: Base address + 880h offset + (4d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MI[31:0]																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CANx_RXIMR*n* field descriptions

Field	Description
31–0 MI[31:0]	Individual Mask Bits Each Individual Mask Bit masks the corresponding bit in both the Mailbox filter and Rx FIFO ID Filter Table element in distinct ways.

CANx_RXIMR*n* field descriptions (continued)

Field	Description
	For Mailbox filters, see the RXMGMASK register description. For Rx FIFO ID Filter Table elements, see the RXFGMASK register description. 0 The corresponding bit in the filter is "don't care." 1 The corresponding bit in the filter is checked.

46.3.20 Memory Error Control Register (CANx_MECR)

This register contains control bits for memory error detection and correction (ECC).

NOTE

When bit CTRL2[ECRWRE] is 0, this register is read-only and writes to this register result in bus transfer errors.

Address: Base address + AE0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ECRWRDIS	0											HANCEI_MSK	FANCEI_MSK	0	CEI_MSK
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HAERRIE	FAERRIE	EXTERRIE	0			RERRDIS	ECCDIS	NCEFAFRZ	0						
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

CANx_MECR field descriptions

Field	Description
31 ECRWRDIS	Error Configuration Register Write Disable Disables writes on this register. 0 Write is enabled. 1 Write is disabled.
30–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 HANCEI_MSK	Host Access With Non-Correctable Errors Interrupt Mask Enables the interrupt in case of non-correctable errors detected in memory reads issued by the host (CPU).

Table continues on the next page...

CANx_MECR field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
18 FANCEI_MSK	FlexCAN Access With Non-Correctable Errors Interrupt Mask Enables the interrupt in case of non-correctable errors detected in memory reads issued by the FlexCAN internal processes. 0 Interrupt is disabled. 1 Interrupt is enabled.
17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 CEI_MSK	Correctable Errors Interrupt Mask Enables the interrupt in case of correctable errors detected in memory reads issued by the host or FlexCAN internal processes. 0 Interrupt is disabled. 1 Interrupt is enabled.
15 HAERRIE	Host Access Error Injection Enable Enables the injection of errors only in memory reads issued by the host (CPU). 0 Injection is disabled. 1 Injection is enabled.
14 FAERRIE	FlexCAN Access Error Injection Enable Enables the injection of errors only in memory reads issued by the FlexCAN internal processes. 0 Injection is disabled. 1 Injection is enabled.
13 EXTERRIE	Extended Error Injection Enable Memory accesses performed by internal FlexCAN processes are 64-bit. This bit extends the error injection on 32-bit memory accesses to the complementary 32-bit word using the same 32-bit error injection data and parity words. See Error Injection Data Pattern Register (FCERRIDPR) and Error Injection Parity Pattern Register (FCERRIPPR) 0 Error injection is applied only to the 32-bit word. 1 Error injection is applied to the 64-bit word.
12–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 RERRDIS	Error Report Disable Disables the update of the error report registers. The update of error-related flags and the generation of bus transfer errors are still active. **NOTE:** When reading the report registers, this bit must be cleared to assure coherence on the consecutive register reads. 0 Enable updates of the error report registers. 1 Disable updates of the error report registers.
8 ECCDIS	Error Correction Disable

Table continues on the next page...

CANx_MECR field descriptions (continued)

Field	Description
	Disables completely the memory detection and correction mechanism. Besides disabling the error report mechanism, it also stops the update of the error-related flags and generation of bus transfer errors. The parity bits continue being calculated and written into memory on write transactions. 0 Enable memory error correction. 1 Disable memory error correction.
7 NCEFAFRZ	Non-Correctable Errors In FlexCAN Access Put Device In Freeze Mode Determines the response when a non-correctable error is detected in a memory read performed by FlexCAN internal processes. The option to enter Freeze mode is a safety measure to prevent corrupted data from being treated as valid by FlexCAN internal processes. 0 Keep normal operation. 1 Put FlexCAN in Freeze mode (according to the "Freeze mode" section).
6–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

46.3.21 Error Injection Address Register (CANx_ERRIAR)

This register holds the address where error is to be injected.

Use the following table to convert from the memory map address to the location in the physical FlexCAN RAM:

RAM contents	Injection address	Memory map
FlexCAN Registers	Not mapped	-
MBs	0x0000	0x0080
Reserved	-	0x0480
RXIMRs	0x0400	0x0880
Reserved	-	0x0980
RXFIR_0	0x0500	0x0A80
RXFIR_1	0x0504	0x0A84
RXFIR_2	0x0508	0x0A88
RXFIR_3	0x050C	0x0A8C
RXFIR_4	0x0510	0x0A90
RXFIR_5	0x0514	0x0A94
Reserved	-	0x0A98
RXMGMASK	0x0520	0x0AA0
RXFGMASK	0x0524	0x0AA4
RX14MASK	0x0528	0x0AA8
RX15MASK	0x052C	0x0AAC
SMB_TX	0x0530	0x0AB0

Table continues on the next page...

RAM contents	Injection address	Memory map
SMB_RX0	0x0540	0x0AC0
SMB_RX1	0x0550	0x0AD0
ECC Registers	Not mapped	0x0AE0
Reserved	-	0x0B00

Address: Base address + AE4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																		INJADDR													
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_ERRIAR field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–0 INJADDR	Address Where Error Is To Be Injected

46.3.22 Error Injection Data Pattern Register (CANx_ERRIDPR)

Holds the error pattern to be injected in the data word read from memory.

Address: Base address + AE8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DFLIP																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_ERRIDPR field descriptions

Field	Description
31–0 DFLIP	Data flip pattern Bits set to 1 in the flip pattern cause the corresponding data bit in the word read from memory to invert.

46.3.23 Error Injection Parity Pattern Register (CANx_ERRIPPR)

Holds the error pattern to be injected in parity bits read from memory along with data word.

Bits set to 1 in the flip pattern cause the corresponding parity bit in the word read from memory to invert.

Address: Base address + AECh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			PFLIP3					0			PFLIP2					0			PFLIP1					0			PFLIP0				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_ERRIPPR field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 PFLIP3	Parity Flip Pattern For Byte 3 (most significant)
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 PFLIP2	Parity Flip Pattern For Byte 2
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 PFLIP1	Parity Flip Pattern For Byte 1
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 PFLIP0	Parity Flip Pattern For Byte 0 (Least Significant)

46.3.24 Error Report Address Register (CANx_RERRAR)

Reports the address used for an access in which an error (correctable or non-correctable) was detected, and also reports the identification of the source of that access.

This address is always reported using a 32-bit alignment. Non-aligned accesses (ERRADDR[1:0] non-0) are reported with the address aligned and data is reported in RERRDR accordingly shifted. In case of errors detected in accesses larger than 32-bit (as performed by FlexCAN internal processes), the address of the 32-bit word in which the error was detected is reported. In case of errors detected in more than one 32-bit word, only the least significant address is reported.

Address: Base address + AF0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							NCE	0					SAID		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		ERRADDR													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_RERRAR field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 NCE	Non-Correctable Error Indicates that the report is due to an non-correctable error. 0 Reporting a correctable-error 1 Reporting a non-correctable error
23–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 SAID	SAID[2] — Identification of the requestor of the memory read request: • 0 = Requested by FlexCAN internal processes • 1 = Requested by Host (CPU) SAID[1] — Details of FlexCAN operation: • 0 = Move • 1 = Scanning SAID[0] — Operation that requested the memory read: • 0 = Transmission • 1 = Reception **Table 46-184. Source of memory access** (see below)
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–0 ERRADDR	Address Where The Error Was Detected See the description of the Error Injection Address Register (ERRIAR).

Table 46-184. Source of memory access

SAID[2:0]	Error during...
0	Tx Move
1	Rx Move
2	Tx Arbitration
3	Rx Match
4	Tx Move
5-7	Reserved

46.3.25 Error Report Data Register (CANx_RERRDR)

Reports the raw data (unmodified by the correction performed by ECC logic) read from memory with error. The value reported does not represent the transient values of the BUSY bit (see the "Message Buffer Code for Rx buffers" table) when reading a Message Buffer.

Address: Base address + AF4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_RERRDR field descriptions

Field	Description
31–0 RDATA	Raw data word read from memory with error

46.3.26 Error Report Syndrome Register (CANx_RERRSYNR)

Holds the syndrome detected in a memory read with error. It also reports the bytes which were read in this 32-bit read transaction.

Each SYNDn field indicates the type of error and which bit in byte (n) is affected by the error. (SYND3 corresponds to the most significant byte in the data word read from memory; SYND0 corresponds to the least significant.)

Table 46-187. Syndrome Definition

SYNDn (hex)	Type	Bit affected
00	-	none (no error)
01	Code	0
02	Code	1
04	Code	2
07	Data	5
08	Code	3
0E	Data	7
10	Code	4
13	Data	2
15	Data	6

Table continues on the next page...

Table 46-187. Syndrome Definition (continued)

SYNDn (hex)	Type	Bit affected
16	Data	1
19	Data	3
1A	Data	4
1C	Data	0
All others	-	Non-correctable error

Each BEn field indicates which byte in the 32-bit word reported was effectively read. The syndrome bits are calculated for all bytes, even for the non-read ones. Errors detected in non-read bytes are indicated (see the "Error Indication" section) and reported (see the "Error Reporting" section).

Address: Base address + AF8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BE3	0		SYND3					BE2	0		SYND2				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BE1	0		SYND1					BE0	0		SYND0				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_RERRSYNR field descriptions

Field	Description
31 BE3	Byte Enabled For Byte 3 (Most Significant) 0 The byte was not read. 1 The byte was read.
30–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 SYND3	Error Syndrome For Byte 3 (Most Significant) See the "Syndrome Definition" table.
23 BE2	Byte Enabled For Byte 2 0 The byte was not read. 1 The byte was read.
22–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 SYND2	Error Syndrome For Byte 2 See the "Syndrome Definition" table.
15 BE1	Byte Enabled For Byte 1

Table continues on the next page...

CANx_RERRSYNR field descriptions (continued)

Field	Description
	0 The byte was not read. 1 The byte was read.
14–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 SYND1	Error Syndrome for Byte 1 See the "Syndrome Definition" table.
7 BE0	Byte Enabled For Byte 0 (least significant) 0 The byte was not read. 1 The byte was read.
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 SYND0	Error Syndrome For Byte 0 (least significant) See the "Syndrome Definition" table.

46.3.27 Error Status Register (CANx_ERRSR)

Holds the status bits of the error correction and detection operations. After raised, these flags must be cleared by writing 1 to them. Writing 0 has no effect.

Address: Base address + AFCh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												HANCEIF	FANCEIF	0	CEIF
W													w1c	w1c		w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												HANCEIOF	FANCEIOF	0	CEIOF
W													w1c	w1c		w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CANx_ERRSR field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 HANCEIF	Host Access With Non-Correctable Error Interrupt Flag Indicates that a non-correctable error was detected in a memory access initiated by Host. A bus transfer error is asserted for that access. If MECR[HANCEI_MSK] is set, the interrupt is asserted. 0 No non-correctable errors were detected in Host accesses so far. 1 A non-correctable error was detected in a Host access.
18 FANCEIF	FlexCAN Access With Non-Correctable Error Interrupt Flag

Table continues on the next page...

CANx_ERRSR field descriptions (continued)

Field	Description
	Indicates that a non-correctable error was detected in a memory access initiated by FlexCAN internal processes. If MECR[FANCEI_MSK] is set, the interrupt is asserted. 0 No non-correctable errors were detected in FlexCAN accesses so far. 1 A non-correctable error was detected in a FlexCAN access.
17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 CEIF	Correctable Error Interrupt Flag Indicates that a correctable error was detected in a memory access. If MECR[CEI_MSK] is set, the interrupt is asserted. 0 No correctable errors were detected so far. 1 A correctable error was detected.
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 HANCEIOF	Host Access With Non-Correctable Error Interrupt Overrun Flag Indicates that a non-correctable error was detected in a memory access initiated by Host when HANCEIF was set. No interrupt is associated to this flag. (See the "Error Indication" section.) 0 No overrun on non-correctable errors in Host access 1 Overrun on non-correctable errors in Host access
2 FANCEIOF	FlexCAN Access With Non-Correctable Error Interrupt Overrun Flag Indicates that a non-correctable error was detected in a memory access initiated by FlexCAN internal processes when FANCEIF was set. No interrupt is associated to this flag. (See the "Error Indication" section.) 0 No overrun on non-correctable errors in FlexCAN access 1 Overrun on non-correctable errors in FlexCAN access
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 CEIOF	Correctable Error Interrupt Overrun Flag Indicates that a correctable error was detected in a memory access when CEIF was set. No interrupt is associated to this flag. (See the "Error Indication" section.) 0 No overrun on correctable errors 1 Overrun on correctable errors

46.3.80 Message buffer structure

The message buffer structure used by the FlexCAN module is represented in the following figure. Both Extended (29-bit identifier) and Standard (11-bit identifier) frames used in the CAN specification (Version 2.0 Part B) are represented. Each individual MB is formed by 16 bytes.

The memory area from 0x80 to 0x47C is used by the mailboxes.

Table 46-282. Message buffer structure

	31 30 29 28	27 24	23	22	21	20	19 18 17 16	15 8	7 0
0x0		CODE		SRR	IDE	RTR	DLC	TIME STAMP	
0x4	PRIO	ID (Standard/Extended)						ID (Extended)	
0x8	Data Byte 0		Data Byte 1					Data Byte 2	Data Byte 3
0xC	Data Byte 4		Data Byte 5					Data Byte 6	Data Byte 7

(shaded) = Unimplemented or Reserved

CODE — Message Buffer Code

This 4-bit field can be accessed (read or write) by the CPU and by the FlexCAN module itself, as part of the message buffer matching and arbitration process. The encoding is shown in Table 46-283 and Table 46-284. See Functional description for additional information.

Table 46-283. Message buffer code for Rx buffers

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
0b0000: INACTIVE — MB is not active.	INACTIVE	—	—	—	MB does not participate in the matching process.
0b0100: EMPTY — MB is active and empty.	EMPTY	—	FULL	—	When a frame is received successfully (after the Move-in) process), the CODE field is automatically updated to FULL.

Table continues on the next page...

Table 46-283. Message buffer code for Rx buffers (continued)

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
0b0010: FULL — MB is full.	FULL	Yes	FULL	—	The act of reading the C/S word followed by unlocking the MB (SRV) does not make the code return to EMPTY. It remains FULL. If a new frame is moved to the MB after the MB was serviced, the code still remains FULL. See Matching process for matching details related to FULL code.
		No	OVERRUN	—	If the MB is FULL and a new frame is moved to this MB before the CPU services it, the CODE field is automatically updated to OVERRUN. See Matching process for details about overrun behavior.
0b0110: OVERRUN — MB is being overwritten into a full buffer.	OVERRUN	Yes	FULL	—	If the CODE field indicates OVERRUN and CPU has serviced the MB, when a new frame is moved to the MB then the code returns to FULL.
		No	OVERRUN	—	If the CODE field already indicates OVERRUN, and another new frame must be moved, the MB will be overwritten again, and the code will remain OVERRUN. See Matching process for details about overrun behavior.

Table continues on the next page...

Table 46-283. Message buffer code for Rx buffers (continued)

CODE description	Rx code BEFORE receive new frame	SRV[1]	Rx code AFTER successful reception[2]	RRS[3]	Comment
0b1010: RANSWER[4] — A frame was configured to recognize a Remote Request Frame and transmit a Response Frame in return.	RANSWER	—	TANSWER(0b1110)	0	A Remote Answer was configured to recognize a remote request frame received. After that an MB is set to transmit a response frame. The code is automatically changed to TANSWER (0b1110). See Matching process for details. If CTRL2[RRS] is negated, transmit a response frame whenever a remote request frame with the same ID is received.
		—	—	1	This code is ignored during matching and arbitration process. See Matching process for details.
CODE[0]=1b1: BUSY — FlexCAN is updating the contents of the MB. The CPU must not access the MB.	BUSY[5]	—	FULL	—	Indicates that the MB is being updated. It will be negated automatically and does not interfere with the next CODE.
		—	OVERRUN	—	

1. SRV: Serviced MB. MB was read and unlocked by reading TIMER or other MB.
2. A frame is considered a successful reception after the frame to be moved to MB (move-in process). See Move-in for details.
3. Remote Request Stored bit from CTRL2 register. See section "Control 2 Register (CTRL2)" for details.
4. Code 0b1010 is not considered Tx and an MB with this code should not be aborted.
5. Note that for Tx MBs, the BUSY bit should be ignored upon read, except when AEN bit is set in the MCR register. If this bit is asserted, the corresponding MB does not participate in the matching process.

Table 46-284. Message buffer code for Tx buffers

CODE Description	Tx Code BEFORE tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
0b1000: INACTIVE — MB is not active	INACTIVE	—	—	MB does not participate in arbitration process.

Table continues on the next page...

Table 46-284. Message buffer code for Tx buffers (continued)

CODE Description	Tx Code BEFORE tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
0b1001: ABORT — MB is aborted	ABORT	—	—	MB does not participate in arbitration process.
0b1100: DATA — MB is a Tx Data Frame (MB RTR must be 0)	DATA	0	INACTIVE	Transmit data frame unconditionally once. After transmission, the MB automatically returns to the INACTIVE state.
0b1100: REMOTE — MB is a Tx Remote Request Frame (MB RTR must be 1)	REMOTE	1	EMPTY	Transmit remote request frame unconditionally once. After transmission, the MB automatically becomes an Rx Empty MB with the same ID.
0b1110: TANSWER — MB is a Tx Response Frame from an incoming Remote Request Frame	TANSWER	—	RANSWER	This is an intermediate code that is automatically written to the MB by the CHI as a result of a match to a remote request frame. The remote response frame will be transmitted unconditionally once, and then the code will automatically return to RANSWER (0b1010). The CPU can also write this code with the same effect. The remote response frame can be either a data frame or another remote request frame depending on the RTR bit value. See Matching process and Arbitration process for details.

SRR — Substitute Remote Request

Fixed recessive bit, used only in extended format. It must be set to one by the user for transmission (Tx Buffers) and will be stored with the value received on the CAN bus for Rx receiving buffers. It can be received as either recessive or dominant. If FlexCAN receives this bit as dominant, then it is interpreted as an arbitration loss.

1 = Recessive value is compulsory for transmission in extended format frames

0 = Dominant is not a valid value for transmission in extended format frames

IDE — ID Extended Bit

This field identifies whether the frame format is standard or extended.

1 = Frame format is extended

0 = Frame format is standard

RTR — Remote Transmission Request

This bit affects the behavior of remote frames and is part of the reception filter. See Table 46-283, Table 46-284, and the description of the RRS bit in Control 2 Register (CTRL2) for additional details.

If FlexCAN transmits this bit as '1' (recessive) and receives it as '0' (dominant), it is interpreted as an arbitration loss. If this bit is transmitted as '0' (dominant), then if it is received as '1' (recessive), the FlexCAN module treats it as a bit error. If the value received matches the value transmitted, it is considered a successful bit transmission.

1 = Indicates the current MB may have a remote request frame to be transmitted if MB is Tx. If the MB is Rx then incoming remote request frames may be stored.

0 = Indicates the current MB has a data frame to be transmitted. In Rx MB it may be considered in matching processes.

DLC — Length of Data in Bytes

This 4-bit field is the length (in bytes) of the Rx or Tx data, which is located in offset 0x8 through 0xF of the MB space (see the Table 46-282). In reception, this field is written by the FlexCAN module, copied from the DLC (Data Length Code) field of the received frame. In transmission, this field is written by the CPU and corresponds to the DLC field value of the frame to be transmitted. When RTR = 1, the frame to be transmitted is a remote frame and does not include the data field, regardless of the DLC field (see Table 46-285).

TIME STAMP — Free-Running Counter Time Stamp

This 16-bit field is a copy of the Free-Running Timer, captured for Tx and Rx frames at the time when the beginning of the Identifier field appears on the CAN bus.

PRIO — Local priority

This 3-bit field is used only when LPRIO_EN bit is set in MCR, and it only makes sense for Tx mailboxes. These bits are not transmitted. They are appended to the regular ID to define the transmission priority. See Arbitration process.

ID — Frame Identifier

In standard frame format, only the 11 most significant bits (28 to 18) are used for frame identification in both receive and transmit cases. The 18 least significant bits are ignored. In extended frame format, all bits are used for frame identification in both receive and transmit cases.

DATA BYTE 0–7 — Data Field

Up to eight bytes can be used for a data frame.

For Rx frames, the data is stored as it is received from the CAN bus. DATA BYTE (*n*) is valid only if *n* is less than DLC as shown in the table below.

For Tx frames, the CPU prepares the data field to be transmitted within the frame.

Table 46-285. DATA BYTEs validity

DLC	Valid DATA BYTEs
0	none
1	DATA BYTE 0
2	DATA BYTE 0–1
3	DATA BYTE 0–2
4	DATA BYTE 0–3
5	DATA BYTE 0–4
6	DATA BYTE 0–5
7	DATA BYTE 0–6
8	DATA BYTE 0–7

46.3.81 Rx FIFO structure

When the MCR[RFEN] bit is set, the memory area from 0x80 to 0xDC (which is normally occupied by MBs 0–5) is used by the reception FIFO engine.

The region 0x80-0x8C contains the output of the FIFO which must be read by the CPU as a message buffer. This output contains the oldest message that has been received but not yet read. The region 0x90-0xDC is reserved for internal use of the FIFO engine.

An additional memory area, which starts at 0xE0 and may extend up to 0x2DC (normally occupied by MBs 6–37) depending on the CTRL2[RFFN] field setting, contains the ID filter table (configurable from 8 to 128 table elements) that specifies filtering criteria for accepting frames into the FIFO.

Out of reset, the ID filter table flexible memory area defaults to 0xE0 and extends only to 0xFC, which corresponds to MBs 6 to 7 for RFFN = 0, for backward compatibility with previous versions of FlexCAN.

The following shows the Rx FIFO data structure.

Table 46-286. Rx FIFO structure

	31	28	24	23	22	21	20	19	18	17	16	15	8	7	0
0x80	(reserved)				SRR	IDE	RTR	DLC				TIME STAMP			
0x84	(reserved)	ID standard								ID extended					
0x88	Data byte 0			Data byte 1								Data byte 2		Data byte 3	
0x8C	Data byte 4			Data byte 5								Data byte 6		Data byte 7	
0x90 to 0xDC	Reserved														
0xE0	ID filter table element 0														
0xE4	ID filter table element 1														
0xE8 to 0x2D4	ID filter table elements 2 to 125														
0x2D8	ID filter table element 126														
0x2DC	ID filter table element 127														
	(reserved) = Unimplemented or reserved														

Each ID filter table element occupies an entire 32-bit word and can be compounded by one, two, or four Identifier Acceptance Filters (IDAF) depending on the MCR[IDAM] field setting. The following figures show the IDAF indexation.

The following table shows the three different formats of the ID table elements. Note that all elements of the table must have the same format. See Rx FIFO for more information.

Table 46-287. ID table structure

Format	31	30	29	24	23	16	15	14	13	8	7	1	0
A	RTR	IDE	RXIDA (standard = 29–19, extended = 29–1)										(reserved)
B	RTR	IDE	RXIDB_0 (standard = 29–19, extended = 29–16)				RTR	IDE	RXIDB_1 (standard = 13–3, extended = 13–0)				
C	RXIDC_0 (std/ext = 31–24)				RXIDC_1 (std/ext = 23–16)		RXIDC_2 (std/ext = 15–8)				RXIDC_3 (std/ext = 7–0)		
	(reserved) = Unimplemented or Reserved												

RTR — Remote Frame

This bit specifies if Remote Frames are accepted into the FIFO if they match the target ID.

1 = Remote Frames can be accepted and data frames are rejected

0 = Remote Frames are rejected and data frames can be accepted

IDE — Extended Frame

Specifies whether extended or standard frames are accepted into the FIFO if they match the target ID.

1 = Extended frames can be accepted and standard frames are rejected

0 = Extended frames are rejected and standard frames can be accepted

RXIDA — Rx Frame Identifier (Format A)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, only the 11 most significant bits (29 to 19) are used for frame identification. In the extended frame format, all bits are used.

RXIDB_0, RXIDB_1 — Rx Frame Identifier (Format B)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, the 11 most significant bits (a full standard ID) (29 to 19 and 13 to 3) are used for frame identification. In the extended frame format, all 14 bits of the field are compared to the 14 most significant bits of the received ID.

RXIDC_0, RXIDC_1, RXIDC_2, RXIDC_3 — Rx Frame Identifier (Format C)

Specifies an ID to be used as acceptance criteria for the FIFO. In both standard and extended frame formats, all 8 bits of the field are compared to the 8 most significant bits of the received ID.

46.4 Functional description

The FlexCAN module is a CAN protocol engine with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system is composed by a set of Message Buffers (MB) that store configuration and control data, time stamp, message ID and data (see Message buffer structure). The memory corresponding to the first 38 MBs can be configured to support a FIFO reception scheme with a powerful ID filtering mechanism, capable of checking incoming frames against a table of IDs (up to 128 extended IDs or 256 standard IDs or 512 8-bit ID slices), with individual mask register for up to 32 ID Filter Table elements. Simultaneous reception through FIFO and mailbox is supported. For mailbox reception, a matching algorithm makes it possible to store received frames only into MBs that have the same ID programmed on its ID field. A masking scheme makes it possible to match the ID programmed on the MB with a range

of IDs on received CAN frames. For transmission, an arbitration algorithm decides the prioritization of MBs to be transmitted based on the message ID (optionally augmented by 3 local priority bits) or the MB ordering.

Before proceeding with the functional description, an important concept must be explained. A Message Buffer is said to be "active" at a given time if it can participate in both the Matching and Arbitration processes. An Rx MB with a 0b0000 code is inactive (refer to Table 46-283). Similarly, a Tx MB with a 0b1000 or 0b1001 code is also inactive (refer to Table 46-284).

46.4.1 Transmit process

To transmit a CAN frame, the CPU must prepare a Message Buffer for transmission by executing the following procedure:

1. Check whether the respective interrupt bit is set and clear it.
2. If the MB is active (transmission pending), write the ABORT code (0b1001) to the CODE field of the Control and Status word to request an abortion of the transmission. Wait for the corresponding IFLAG to be asserted by polling the IFLAG register or by the interrupt request if enabled by the respective IMASK. Then read back the CODE field to check if the transmission was aborted or transmitted (see Transmission abort mechanism). If backwards compatibility is desired (MCR[AEN] bit is negated), just write the INACTIVE code (0b1000) to the CODE field to inactivate the MB but then the pending frame may be transmitted without notification (see Mailbox inactivation).
3. Write the ID word.
4. Write the data bytes.
5. Write the DLC, Control, and CODE fields of the Control and Status word to activate the MB.

When the MB is activated, it will participate into the arbitration process and eventually be transmitted according to its priority. At the end of the successful transmission, the value of the Free Running Timer is written into the Time Stamp field, the CODE field in the Control and Status word is updated, the CRC Register is updated, a status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit. The new CODE field after transmission depends on the code that was used to activate the MB (see Table 46-283 and Table 46-284 in Message buffer structure).

When the Abort feature is enabled (MCR[AEN] is asserted), after the Interrupt Flag is asserted for a Mailbox configured as transmit buffer, the Mailbox is blocked, therefore the CPU is not able to update it until the Interrupt Flag is negated by CPU. This means that the CPU must clear the corresponding IFLAG before starting to prepare this MB for a new transmission or reception.

46.4.2 Arbitration process

The arbitration process scans the Mailboxes searching the Tx one that holds the message to be sent in the next opportunity. This Mailbox is called the *arbitration winner*.

The scan starts from the lowest number Mailbox and runs toward the higher ones.

The arbitration process is triggered in the following events:

- From the CRC field of the CAN frame. The start point depends on the CTRL2[TASD] field value.
- During the Error Delimiter field of a CAN frame.
- During the Overload Delimiter field of a CAN frame.
- When the winner is inactivated and the CAN bus has still not reached the first bit of the Intermission field.
- When there is CPU write to the C/S word of a winner MB and the CAN bus has still not reached the first bit of the Intermission field.
- When CHI is in Idle state and the CPU writes to the C/S word of any MB.
- When FlexCAN exits Bus Off state.
- Upon leaving Freeze mode or Low Power mode.

If the arbitration process does not manage to evaluate all Mailboxes before the CAN bus has reached the first bit of the Intermission field the temporary arbitration winner is invalidated and the FlexCAN will not compete for the CAN bus in the next opportunity.

The arbitration process selects the winner among the active Tx Mailboxes at the end of the scan according to both CTRL1[LBUF] and MCR[LPRIOEN] bits settings.

46.4.2.1 Lowest-number Mailbox first

If CTRL1[LBUF] bit is asserted the first (lowest number) active Tx Mailbox found is the arbitration winner. MCR[LPRIOEN] bit has no effect when CTRL1[LBUF] is asserted.

46.4.2.2 Highest-priority Mailbox first

If CTRL1[LBUF] bit is negated, then the arbitration process searches the active Tx Mailbox with the highest priority, which means that this Mailbox's frame would have a higher probability to win the arbitration on CAN bus when multiple external nodes compete for the bus at the same time.

The sequence of bits considered for this arbitration is called the *arbitration value* of the Mailbox. The highest-priority Tx Mailbox is the one that has the lowest arbitration value among all Tx Mailboxes.

If two or more Mailboxes have equivalent arbitration values, the Mailbox with the lowest number is the arbitration winner.

The composition of the arbitration value depends on MCR[LPRIOEN] bit setting.

46.4.2.2.1 Local Priority disabled

If MCR[LPRIOEN] bit is negated the arbitration value is built in the exact sequence of bits as they would be transmitted in a CAN frame (see the following table) in such a way that the Local Priority is disabled.

Table 46-288. Composition of the arbitration value when Local Priority is disabled

Format	Mailbox Arbitration Value (32 bits)				
Standard (IDE = 0)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

46.4.2.2.2 Local Priority enabled

If Local Priority is desired MCR[LPRIOEN] must be asserted. In this case the Mailbox PRIO field is included at the very left of the arbitration value (see the following table).

Table 46-289. Composition of the arbitration value when Local Priority is enabled

Format	Mailbox Arbitration Value (35 bits)					
Standard (IDE = 0)	PRIO (3 bits)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bit)
Extended (IDE = 1)	PRIO (3 bits)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

As the PRIO field is the most significant part of the arbitration value Mailboxes with low PRIO values have higher priority than Mailboxes with high PRIO values regardless the rest of their arbitration values.

Note that the PRIO field is not part of the frame on the CAN bus. Its purpose is only to affect the internal arbitration process.

46.4.2.3 Arbitration process (continued)

After the arbitration winner is found, its content is copied to a hidden auxiliary MB called Tx Serial Message Buffer (Tx SMB), which has the same structure as a normal MB but is not user accessible. This operation is called move-out and after it is done, write access to the corresponding MB is blocked (if the AEN bit in MCR is asserted). The write access is released in the following events:

- After the MB is transmitted
- FlexCAN enters in Freeze mode or Bus Off
- FlexCAN loses the bus arbitration or there is an error during the transmission

At the first opportunity window on the CAN bus, the message on the Tx SMB is transmitted according to the CAN protocol rules. FlexCAN transmits up to eight data bytes, even if the Data Length Code (DLC) field value is greater than that.

Arbitration process can be triggered in the following situations:

- During Rx and Tx frames from CAN CRC field to end of frame. TASD value may be changed to optimize the arbitration start point.
- During CAN BusOff state from TX_ERR_CNT=124 to 128. TASD value may be changed to optimize the arbitration start point.
- During C/S write by CPU in BusIdle. First C/S write starts arbitration process and a second C/S write during this same arbitration restarts the process. If other C/S writes are performed, Tx arbitration process is pending. If there is no arbitration winner after arbitration process has finished, then TX arbitration machine begins a new arbitration process.

  - If there is a pending arbitration and BusIdle state starts then an arbitration process is triggered. In this case the first and second C/S write in BusIdle will not restart the arbitration process. It is possible that there is not enough time to finish arbitration in WaitForBusIdle state and the next state is Idle. In this case the scan is not interrupted, and it is completed during BusIdle state. During this arbitration C/S write does not cause arbitration restart.
- Arbitration winner deactivation during a valid arbitration window.
- Upon Leave Freeze mode (first bit of the WaitForBusIdle state). If there is a re-synchronization during WaitForBusIdle arbitration process is restarted.

Arbitration process stops in the following situation:

- All Mailboxes were scanned
- A Tx active Mailbox is found in case of Lowest Buffer feature enabled
- Arbitration winner inactivation or abort during any arbitration process
- There was not enough time to finish Tx arbitration process (for instance, when a deactivation was performed near the end of frame). In this case arbitration process is pending.
- Error or Overload flag in the bus
- Low Power or Freeze mode request in Idle state

Arbitration is considered pending as described below:

- It was not possible to finish arbitration process in time
- C/S write during arbitration if write is performed in a MB whose number is lower than the Tx arbitration pointer
- Any C/S write if there is no Tx Arbitration process in progress
- Rx Match has just updated a Rx Code to Tx Code
- Entering Busoff state

C/S write during arbitration has the following effect:

- If C/S write is performed in the arbitration winner, a new process is restarted immediately.
- If C/S write is performed in a MB whose number is higher than the Tx arbitration pointer, the ongoing arbitration process will scan this MB as normal.

46.4.3 Receive process

To be able to receive CAN frames into a Mailbox, the CPU must prepare it for reception by executing the following steps:

1. If the Mailbox is active (either Tx or Rx) inactivate the Mailbox (see Mailbox inactivation), preferably with a safe inactivation (see Transmission abort mechanism).
2. Write the ID word
3. Write the EMPTY code (0b0100) to the CODE field of the Control and Status word to activate the Mailbox.

After the MB is activated, it will be able to receive frames that match the programmed filter. At the end of a successful reception, the Mailbox is updated by the *move-in* process (see Move-in) as follows:

1. The received Data field (8 bytes at most) is stored.
2. The received Identifier field is stored.
3. The value of the Free Running Timer at the time of the second bit of frame's Identifier field is written into the Mailbox's Time Stamp field.
4. The received SRR, IDE, RTR, and DLC fields are stored.
5. The CODE field in the Control and Status word is updated (see Table 46-283 and Table 46-284 in Section Message buffer structure).
6. A status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The recommended way for CPU servicing (read) the frame received in an Mailbox is using the following procedure:

1. Read the Control and Status word of that Mailbox.
2. Check if the BUSY bit is deasserted, indicating that the Mailbox is locked. Repeat step 1) while it is asserted. See Mailbox lock mechanism.
3. Read the contents of the Mailbox. Once Mailbox is locked now, its contents won't be modified by FlexCAN Move-in processes. See Move-in.
4. Acknowledge the proper flag at IFLAG registers.
5. Read the Free Running Timer. It is optional but recommended to unlock Mailbox as soon as possible and make it available for reception.

The CPU should synchronize to frame reception by the status flag bit for the specific Mailbox in one of the IFLAG Registers and not by the CODE field of that Mailbox. Polling the CODE field does not work because once a frame was received and the CPU

services the Mailbox (by reading the C/S word followed by unlocking the Mailbox), the CODE field will not return to EMPTY. It will remain FULL, as explained in Table 46-283 . If the CPU tries to workaround this behavior by writing to the C/S word to force an EMPTY code after reading the Mailbox without a prior *safe inactivation*, a newly received frame matching the filter of that Mailbox may be lost.

CAUTION

In summary: *never do polling by reading directly the C/S word of the Mailboxes. Instead, read the IFLAG registers.*

Note that the received frame's Identifier field is always stored in the matching Mailbox, thus the contents of the ID field in an Mailbox may change if the match was due to masking. Note also that FlexCAN does receive frames transmitted by itself if there exists a matching Rx Mailbox, provided the MCR[SRXDIS] bit is not asserted. If the MCR[SRXDIS] bit is asserted, FlexCAN will not store frames transmitted by itself in any MB, even if it contains a matching MB, and no interrupt flag or interrupt signal will be generated due to the frame reception.

To be able to receive CAN frames through the Rx FIFO, the CPU must enable and configure the Rx FIFO during Freeze mode (see Rx FIFO). Upon receiving the Frames Available in Rx FIFO interrupt (see the description of the IFLAG[BUF5I] "Frames available in Rx FIFO" bit in the IMASK1 register), the CPU should service the received frame using the following procedure:

1. Read the Control and Status word (optional – needed only if a mask was used for IDE and RTR bits)
2. Read the ID field (optional – needed only if a mask was used)
3. Read the Data field
4. Read the RXFIR register (optional)
5. Clear the Frames Available in Rx FIFO interrupt by writing 1 to IFLAG[BUF5I] bit (mandatory – releases the MB and allows the CPU to read the next Rx FIFO entry)

46.4.4 Matching process

The matching process scans the MB memory looking for Rx MBs programmed with the same ID as the one received from the CAN bus. If the FIFO is enabled, the priority of scanning can be selected between Mailboxes and FIFO filters. In any case, the matching starts from the lowest number Message Buffer toward the higher ones. If no match is

found within the first structure then the other is scanned subsequently. In the event that the FIFO is full, the matching algorithm will always look for a matching MB outside the FIFO region.

As the frame is being received, it is stored in a hidden auxiliary MB called Rx Serial Message Buffer (Rx SMB).

The matching process start point depends on the following conditions:

- If the received frame is a remote frame, the start point is the CRC field of the frame
- If the received frame is a data frame with DLC field equal to zero, the start point is the CRC field of the frame
- If the received frame is a data frame with DLC field different than zero, the start point is the DATA field of the frame

If a matching ID is found in the FIFO table or in one of the Mailboxes, the contents of the SMB will be transferred to the FIFO or to the matched Mailbox by the move-in process. If any CAN protocol error is detected then no match results will be transferred to the FIFO or to the matched Mailbox at the end of reception.

The matching process scans all matching elements of both Rx FIFO (if enabled) and active Rx Mailboxes (CODE is EMPTY, FULL, OVERRUN or RANSWER) in search of a successful comparison with the matching elements of the Rx SMB that is receiving the frame on the CAN bus. The SMB has the same structure of a Mailbox. The reception structures (Rx FIFO or Mailboxes) associated with the matching elements that had a successful comparison are the *matched structures*. The *matching winner* is selected at the end of the scan among those matched structures and depends on conditions described ahead. See the following table.

Table 46-290. Matching architecture

Structure	SMB[RTR]	CTRL2[RRS]	CTRL2[EACEN]	MB[IDE]	MB[RTR]	MB[ID[1]]	MB[CODE]
Mailbox	0	-	0	cmp[2]	no_cmp[3]	cmp_msk[4]	EMPTY or FULL or OVERRUN
Mailbox	0	-	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	0	-	cmp	no_cmp	cmp	RANSWER
Mailbox	1	1	0	cmp	no_cmp	cmp_msk	EMPTY or FULL or OVERRUN

Table continues on the next page...

Table 46-290. Matching architecture (continued)

Structure	SMB[RTR]	CTRL2[RRS]	CTRL2[EACEN]	MB[IDE]	MB[RTR]	MB[ID[1]]	MB[CODE]
Mailbox	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN
FIFO[5]	-	-	-	cmp_msk	cmp_msk	cmp_msk	-

1. For Mailbox structure, If SMB[IDE] is asserted, the ID is 29 bits (ID Standard + ID Extended). If SMB[IDE] is negated, the ID is only 11 bits (ID Standard). For FIFO structure, the ID depends on IDAM.
2. cmp: Compares the SMB contents with the MB contents regardless the masks.
3. no_cmp: The SMB contents are not compared with the MB contents
4. cmp_msk: Compares the SMB contents with MB contents taking into account the masks.
5. SMB[IDE] and SMB[RTR] are not taken into account when IDAM is type C.

A reception structure is *free-to-receive* when any of the following conditions is satisfied:

- The CODE field of the Mailbox is EMPTY
- The CODE field of the Mailbox is either FULL or OVERRUN and it has already been serviced (the C/S word was read by the CPU and unlocked as described in Mailbox lock mechanism)
- The CODE field of the Mailbox is either FULL or OVERRUN and an inactivation (see Mailbox inactivation) is performed
- The Rx FIFO is not full

The scan order for Mailboxes and Rx FIFO is from the matching element with lowest number to the higher ones.

The matching winner search for Mailboxes is affected by the MCR[IRMQ] bit. If it is negated the matching winner is the first matched Mailbox regardless if it is free-to-receive or not. If it is asserted, the matching winner is selected according to the priority below:

1. the first free-to-receive matched Mailbox;
2. the last non free-to-receive matched Mailbox.

It is possible to select the priority of scan between Mailboxes and Rx FIFO by the CTRL2[MRP] bit.

If the selected priority is Rx FIFO first:

- If the Rx FIFO is a matched structure and is free-to-receive then the Rx FIFO is the matching winner regardless of the scan for Mailboxes
- Otherwise (the Rx FIFO is not a matched structure or is not free-to-receive), then the matching winner is searched among Mailboxes as described above

If the selected priority is Mailboxes first:

- If a free-to-receive matched Mailbox is found, it is the matching winner regardless the scan for Rx FIFO
- If no matched Mailbox is found, then the matching winner is searched in the scan for the Rx FIFO
- If both conditions above are not satisfied and a non free-to-receive matched Mailbox is found then the matching winner determination is conditioned by the MCR[IRMQ] bit:
    - If MCR[IRMQ] bit is negated the matching winner is the first matched Mailbox
    - If MCR[IRMQ] bit is asserted the matching winner is the Rx FIFO if it is a free-to-receive matched structure, otherwise the matching winner is the last non free-to-receive matched Mailbox

See the following table for a summary of matching possibilities.

Table 46-291. Matching possibilities and resulting reception structures

RFEN	IRMQ	MRP	Matched in MB	Matched in FIFO	Reception structure	Description
No FIFO, only MB, match is always MB first						
0	0	X[1]	None[2]	-[3]	None	Frame lost by no match
0	0	X	Free[4]	-	FirstMB	
0	1	X	None	-	None	Frame lost by no match
0	1	X	Free	-	FirstMb	
0	1	X	NotFree	-	LastMB	Overrun
FIFO enabled, no match in FIFO is as if FIFO does not exist						
1	0	X	None	None[5]	None	Frame lost by no match
1	0	X	Free	None	FirstMB	
1	1	X	None	None	None	Frame lost by no match
1	1	X	Free	None	FirstMb	
1	1	X	NotFree	None	LastMB	Overrun
FIFO enabled, Queue disabled						
1	0	0	X	NotFull[6]	FIFO	
1	0	0	None	Full[7]	None	Frame lost by FIFO full (FIFO Overflow)
1	0	0	Free	Full	FirstMB	
1	0	0	NotFree	Full	FirstMB	
1	0	1	None	NotFull	FIFO	
1	0	1	None	Full	None	Frame lost by FIFO full (FIFO Overflow)

Table continues on the next page...

Table 46-291. Matching possibilities and resulting reception structures (continued)

RFEN	IRMQ	MRP	Matched in MB	Matched in FIFO	Reception structure	Description
1	0	1	Free	X	FirstMB	
1	0	1	NotFree	X	FirtsMb	Overrun
FIFO enabled, Queue enabled						
1	1	0	X	NotFull	FIFO	
1	1	0	None	Full	None	Frame lost by FIFO full (FIFO Overflow)
1	1	0	Free	Full	FirstMB	
1	1	0	NotFree	Full	LastMb	Overrun
1	1	1	None	NotFull	FIFO	
1	1	1	Free	X	FirstMB	
1	1	1	NotFree	NotFull	FIFO	
1	1	1	NotFree	Full	LastMb	Overrun

1. This is a don't care condition.
2. Matched in MB "None" means that the frame has not matched any MB (free-to-receive or non-free-to-receive).
3. This is a forbidden condition.
4. Matched in MB "Free" means that the frame matched at least one MB free-to-receive regardless of whether it has matched MBs non-free-to-receive.
5. Matched in FIFO "None" means that the frame has not matched any filter in FIFO. It is as if the FIFO didn't exist (CTRL2[RFEN]=0).
6. Matched in FIFO "NotFull" means that the frame has matched a FIFO filter and has empty slots to receive it.
7. Matched in FIFO "Full" means that the frame has matched a FIFO filter but couldn't store it because it has no empty slots to receive it.

If a non-safe Mailbox inactivation (see Mailbox inactivation) occurs during matching process and the Mailbox inactivated is the temporary matching winner then the temporary matching winner is invalidated. The matching elements scan is not stopped nor restarted, it continues normally. The consequence is that the current matching process works as if the matching elements compared before the inactivation did not exist, therefore a message may be lost.

Suppose, for example, that the FIFO is disabled, IRMQ is enabled and there are two MBs with the same ID, and FlexCAN starts receiving messages with that ID. Let us say that these MBs are the second and the fifth in the array. When the first message arrives, the matching algorithm will find the first match in MB number 2. The code of this MB is EMPTY, so the message is stored there. When the second message arrives, the matching algorithm will find MB number 2 again, but it is not "free-to-receive", so it will keep looking and find MB number 5 and store the message there. If yet another message with the same ID arrives, the matching algorithm finds out that there are no matching MBs that are "free-to-receive", so it decides to overwrite the last matched MB, which is number 5. In doing so, it sets the CODE field of the MB to indicate OVERRUN.

The ability to match the same ID in more than one MB can be exploited to implement a reception queue (in addition to the full featured FIFO) to allow more time for the CPU to service the MBs. By programming more than one MB with the same ID, received messages will be queued into the MBs. The CPU can examine the Time Stamp field of the MBs to determine the order in which the messages arrived.

Matching to a range of IDs is possible by using ID Acceptance Masks. FlexCAN supports individual masking per MB. Refer to the description of the Rx Individual Mask Registers (RXIMRx). During the matching algorithm, if a mask bit is asserted, then the corresponding ID bit is compared. If the mask bit is negated, the corresponding ID bit is "don't care". Please note that the Individual Mask Registers are implemented in RAM, so they are not initialized out of reset. Also, they can only be programmed while the module is in Freeze mode; otherwise, they are blocked by hardware.

FlexCAN also supports an alternate masking scheme with only four mask registers (RGXMASK, RX14MASK, RX15MASK and RXFGMASK) for backwards compatibility with legacy applications. This alternate masking scheme is enabled when the IRMQ bit in the MCR Register is negated.

46.4.5 Move process

There are two types of move process: move-in and move-out.

46.4.5.1 Move-in

The move-in process is the copy of a message received by an Rx SMB to a Rx Mailbox or FIFO that has matched it. If the move destination is the Rx FIFO, attributes of the message are also copied to the RXFIR FIFO. Each Rx SMB has its own move-in process, but only one is performed at a given time as described ahead. The move-in starts only when the message held by the Rx SMB has a corresponding matching winner (see Matching process) and all of the following conditions are true:

- The CAN bus has reached or let past either:
  - The second bit of Intermission field next to the frame that carried the message that is in the Rx SMB
  - The first bit of an overload frame next to the frame that carried the message that is in the Rx SMB
- There is no ongoing matching process

- The destination Mailbox is not locked by the CPU
- There is no ongoing move-in process from another Rx SMB. If more than one move-in processes are to be started at the same time both are performed and the newest substitutes the oldest.

The term *pending move-in* is used throughout the documentation and stands for a move-to-be that still does not satisfy all of the aforementioned conditions.

The move-in is cancelled and the Rx SMB is able to receive another message if any of the following conditions is satisfied:

- The destination Mailbox is inactivated after the CAN bus has reached the first bit of Intermission field next to the frame that carried the message and its matching process has finished
- There is a previous pending move-in to the same destination Mailbox
- The Rx SMB is receiving a frame transmitted by the FlexCAN itself and the self-reception is disabled (MCR[SRXDIS] bit is asserted)
- Any CAN protocol error is detected

Note that the pending move-in is not cancelled if the module enters Freeze or Low-Power mode. It only stays on hold waiting for exiting Freeze and Low-Power mode and to be unlocked. If an MB is unlocked during Freeze mode, the move-in happens immediately.

The move-in process is the execution by the FlexCAN of the following steps:

1. if the message is destined to the Rx FIFO, push IDHIT into the RXFIR FIFO;
2. reads the words DATA0-3 and DATA4-7 from the Rx SMB;
3. writes it in the words DATA0-3 and DATA4-7 of the Rx Mailbox;
4. reads the words Control/Status and ID from the Rx SMB;
5. writes it in the words Control/Status and ID of the Rx Mailbox, updating the CODE field.

The move-in process is not atomic, in such a way that it is immediately cancelled by the inactivation of the destination Mailbox (see Mailbox inactivation) and in this case the Mailbox may be left partially updated, thus incoherent. The exception is if the move-in destination is an Rx FIFO Message Buffer, then the process cannot be cancelled.

The BUSY Bit (least significant bit of the CODE field) of the destination Message Buffer is asserted while the move-in is being performed to alert the CPU that the Message Buffer content is temporarily incoherent.

46.4.5.2 Move-out

The move-out process is the copy of the content from a Tx Mailbox to the Tx SMB when a message for transmission is available (see Section "Arbitration process"). The move-out occurs in the following conditions:

- The first bit of Intermission field
- During Bus Off state when TX Error Counter is in the 124 to 128 range
- During Bus Idle state
- During Wait For Bus Idle state

The move-out process is not atomic. Only the CPU has priority to access the memory concurrently out of Bus Idle state. In Bus Idle, the move-out has the lowest priority to the concurrent memory accesses.

46.4.6 Data coherence

In order to maintain data coherency and FlexCAN proper operation, the CPU must obey the rules described in Transmit process and Receive process.

46.4.6.1 Transmission abort mechanism

The abort mechanism provides a safe way to request the abortion of a pending transmission. A feedback mechanism is provided to inform the CPU if the transmission was aborted or if the frame could not be aborted and was transmitted instead.

Two primary conditions must be fulfilled in order to abort a transmission:

- MCR[AEN] bit must be asserted
- The first CPU action must be the writing of abort code (0b1001) into the CODE field of the Control and Status word.

The active MBs configured as transmission must be aborted first and then they may be updated. If the abort code is written to a Mailbox that is currently being transmitted, or to a Mailbox that was already loaded into the SMB for transmission, the write operation is blocked and the MB is kept active, but the abort request is captured and kept pending until one of the following conditions are satisfied:

- The module loses the bus arbitration
- There is an error during the transmission
- The module is put into Freeze mode

- The module enters in BusOff state
- There is an overload frame

If none of the conditions above are reached, the MB is transmitted correctly, the interrupt flag is set in the IFLAG register, and an interrupt to the CPU is generated (if enabled). The abort request is automatically cleared when the interrupt flag is set. On the other hand, if one of the above conditions is reached, the frame is not transmitted; therefore, the abort code is written into the CODE field, the interrupt flag is set in the IFLAG, and an interrupt is (optionally) generated to the CPU.

If the CPU writes the abort code before the transmission begins internally, then the write operation is not blocked; therefore, the MB is updated and the interrupt flag is set. In this way the CPU just needs to read the abort code to make sure the active MB was *safely inactivated*. Although the AEN bit is asserted and the CPU wrote the abort code, in this case the MB is inactivated and not aborted, because the transmission did not start yet. One Mailbox is only aborted when the abort request is captured and kept pending until one of the previous conditions are satisfied.

The abort procedure can be summarized as follows:

- CPU checks the corresponding IFLAG and clears it, if asserted.
- CPU writes 0b1001 into the CODE field of the C/S word.
- CPU waits for the corresponding IFLAG indicating that the frame was either transmitted or aborted.
- CPU reads the CODE field to check if the frame was either transmitted (CODE=0b1000) or aborted (CODE=0b1001).
- It is necessary to clear the corresponding IFLAG in order to allow the MB to be reconfigured.

46.4.6.2 Mailbox inactivation

Inactivation is a mechanism provided to protect the Mailbox against updates by the FlexCAN internal processes, thus allowing the CPU to rely on Mailbox data coherence after having updated it, even in Normal mode.

Inactivation of transmission Mailboxes must be performed just when MCR[AEN] bit is deasserted.

If a Mailbox is inactivated, it participates in neither the arbitration process nor the matching process until it is reactivated. See Transmit process and Receive process for more detailed instruction on how to inactivate and reactivate a Mailbox.

To inactivate a Mailbox, the CPU must update its CODE field to INACTIVE (either 0b0000 or 0b1000).

Because the user is not able to synchronize the CODE field update with the FlexCAN internal processes, an inactivation can lead to undesirable results:

- A frame in the bus that matches the filtering of the inactivated Rx Mailbox may be lost without notice, even if there are other Mailboxes with the same filter
- A frame containing the message within the inactivated Tx Mailbox may be transmitted without notice

In order to eliminate such risk and perform a *safe inactivation* the CPU must use the following mechanism along with the inactivation itself:

- For Tx Mailboxes, the Transmission Abort (see Transmission abort mechanism)

The inactivation automatically unlocks the Mailbox (see Mailbox lock mechanism).

NOTE

Message Buffers that are part of the Rx FIFO cannot be inactivated. There is no write protection on the FIFO region by FlexCAN. CPU must maintain data coherency in the FIFO region when RFEN is asserted.

46.4.6.3 Mailbox lock mechanism

Other than Mailbox inactivation, FlexCAN has another data coherence mechanism for the receive process. When the CPU reads the Control and Status word of an Rx MB with codes FULL or OVERRUN, FlexCAN assumes that the CPU wants to read the whole MB in an atomic operation, and therefore it sets an internal lock flag for that MB. The lock is released when the CPU reads the Free Running Timer (global unlock operation), or when it reads the Control and Status word of another MB regardless of its code. A CPU write into C/S word also unlocks the MB, but this procedure is not recommended for normal unlock use because it cancels a pending-move and potentially may lose a received message. The MB locking is done to prevent a new frame to be written into the MB while the CPU is reading it.

NOTE

The locking mechanism applies only to Rx MBs that are not part of FIFO and have a code different than INACTIVE

(0b0000) or EMPTY[1] (0b0100). Also, Tx MBs can not be locked.

Suppose, for example, that the FIFO is disabled and the second and the fifth MBs of the array are programmed with the same ID, and FlexCAN has already received and stored messages into these two MBs. Suppose now that the CPU decides to read MB number 5 and at the same time another message with the same ID is arriving. When the CPU reads the Control and Status word of MB number 5, this MB is locked. The new message arrives and the matching algorithm finds out that there are no "free-to-receive" MBs, so it decides to override MB number 5. However, this MB is locked, so the new message can not be written there. It will remain in the SMB waiting for the MB to be unlocked, and only then will be written to the MB.

If the MB is not unlocked in time and yet another new message with the same ID arrives, then the new message overwrites the one on the SMB and there will be no indication of lost messages either in the CODE field of the MB or in the Error and Status Register.

While the message is being moved-in from the SMB to the MB, the BUSY bit on the CODE field is asserted. If the CPU reads the Control and Status word and finds out that the BUSY bit is set, it should defer accessing the MB until the BUSY bit is negated.

Note

If the BUSY bit is asserted or if the MB is empty, then reading the Control and Status word does not lock the MB.

Inactivation takes precedence over locking. If the CPU inactivates a locked Rx MB, then its lock status is negated and the MB is marked as invalid for the current matching round. Any pending message on the SMB will not be transferred anymore to the MB. An MB is unlocked when the CPU reads the Free Running Timer Register (see Section "Free Running Timer Register (TIMER)"), or the C/S word of another MB.

Lock and unlock mechanisms have the same functionality in both Normal and Freeze modes.

An unlock during Normal or Freeze mode results in the move-in of the pending message. However, the move-in is postponed if an unlock occurs during a low power mode (see Modes of operation) and it will take place only when the module resumes to Normal or Freeze modes.

1. In previous FlexCAN versions, reading the C/S word locked the MB even if it was EMPTY. This behavior is maintained when the IRMQ bit is negated.

46.4.7 Rx FIFO

The Rx FIFO is receive-only and is enabled by asserting the MCR[RFEN] bit. The reset value of this bit is zero to maintain software backward compatibility with previous versions of the module that did not have the FIFO feature. The FIFO is 6-message deep. The memory region occupied by the FIFO structure (both Message Buffers and FIFO engine) is described in Rx FIFO structure. The CPU can read the received messages sequentially, in the order they were received, by repeatedly reading a Message Buffer structure at the output of the FIFO.

The IFLAG[BUF5I] (Frames available in Rx FIFO) is asserted when there is at least one frame available to be read from the FIFO. An interrupt is generated if it is enabled by the corresponding mask bit. Upon receiving the interrupt, the CPU can read the message (accessing the output of the FIFO as a Message Buffer) and the RXFIR register and then clear the interrupt. If there are more messages in the FIFO the act of clearing the interrupt updates the output of the FIFO with the next message and update the RXFIR with the attributes of that message, reissuing the interrupt to the CPU. Otherwise, the flag remains negated. The output of the FIFO is only valid whilst the IFLAG[BUF5I] is asserted.

The IFLAG[BUF6I] (Rx FIFO Warning) is asserted when the number of unread messages within the Rx FIFO is increased to 5 from 4 due to the reception of a new one, meaning that the Rx FIFO is almost full. The flag remains asserted until the CPU clears it.

The IFLAG[BUF7I] (Rx FIFO Overflow) is asserted when an incoming message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox. The flag remains asserted until the CPU clears it.

Clearing one of those three flags does not affect the state of the other two.

An interrupt is generated if an IFLAG bit is asserted and the corresponding mask bit is asserted too.

A powerful filtering scheme is provided to accept only frames intended for the target application, reducing the interrupt servicing work load. The filtering criteria is specified by programming a table of up to 128 32-bit registers, according to CTRL2[RFFN] setting, that can be configured to one of the following formats (see also Rx FIFO structure):

- Format A: 128 IDAFs (extended or standard IDs including IDE and RTR)
- Format B: 256 IDAFs (standard IDs or extended 14-bit ID slices including IDE and RTR)
- Format C: 512 IDAFs (standard or extended 8-bit ID slices)

Note

A chosen format is applied to all entries of the filter table. It is not possible to mix formats within the table.

Every frame available in the FIFO has a corresponding IDHIT (Identifier Acceptance Filter Hit Indicator) that can be read by accessing the RXFIR register. The RXFIR[IDHIT] field refers to the message at the output of the FIFO and is valid while the IFLAG[BUF5I] flag is asserted. The RXFIR register must be read only before clearing the flag, which guarantees that the information refers to the correct frame within the FIFO.

Up to 32 elements of the filter table are individually affected by the Individual Mask Registers (RXIMRx), according to the setting of CTRL2[RFFN], allowing very powerful filtering criteria to be defined. If the IRMQ bit is negated, then the FIFO filter table is affected by RXFGMASK.

46.4.8 CAN protocol related features

This section describes the CAN protocol related features.

46.4.8.1 Remote frames

Remote frame is a special kind of frame. The user can program a mailbox to be a Remote Request Frame by writing the mailbox as Transmit with the RTR bit set to '1'. After the remote request frame is transmitted successfully, the mailbox becomes a Receive Message Buffer, with the same ID as before.

When a remote request frame is received by FlexCAN, it can be treated in three ways, depending on Remote Request Storing (CTRL2[RRS]) and Rx FIFO Enable (MCR[RFEN]) bits:

- If RRS is negated the frame's ID is compared to the IDs of the Transmit Message Buffers with the CODE field 0b1010. If there is a matching ID, then this mailbox frame will be transmitted. Note that if the matching mailbox has the RTR bit set, then FlexCAN will transmit a remote frame as a response. The received remote request frame is not stored in a receive buffer. It is only used to trigger a transmission of a frame in response. The mask registers are not used in remote frame matching, and all ID bits (except RTR) of the incoming received frame should match. In the case that a remote request frame was received and matched a mailbox, this message buffer

immediately enters the internal arbitration process, but is considered as normal Tx mailbox, with no higher priority. The data length of this frame is independent of the DLC field in the remote frame that initiated its transmission.

- If RRS is asserted the frame's ID is compared to the IDs of the receive mailboxes with the CODE field 0b0100, 0b0010 or 0b0110. If there is a matching ID, then this mailbox will store the remote frame in the same fashion of a data frame. No automatic remote response frame will be generated. The mask registers are used in the matching process.
- If RFEN is asserted FlexCAN will not generate an automatic response for remote request frames that match the FIFO filtering criteria. If the remote frame matches one of the target IDs, it will be stored in the FIFO and presented to the CPU. Note that for filtering formats A and B, it is possible to select whether remote frames are accepted or not. For format C, remote frames are always accepted (if they match the ID). Remote Request Frames are considered as normal frames, and generate a FIFO overflow when a successful reception occurs and the FIFO is already full.

46.4.8.2 Overload frames

FlexCAN does transmit overload frames due to detection of following conditions on CAN bus:

- Detection of a dominant bit in the first/second bit of Intermission
- Detection of a dominant bit at the 7th bit (last) of End of Frame field (Rx frames)
- Detection of a dominant bit at the 8th bit (last) of Error Frame Delimiter or Overload Frame Delimiter

46.4.8.3 Time stamp

The value of the Free Running Timer is sampled at the beginning of the Identifier field on the CAN bus, and is stored at the end of "move-in" in the TIME STAMP field, providing network behavior with respect to time.

The Free Running Timer can be reset upon a specific frame reception, enabling network time synchronization. See the TSYN description in the description of the Control 1 Register (CTRL1).

46.4.8.4 Protocol timing

The following figure shows the structure of the clock generation circuitry that feeds the CAN Protocol Engine (PE) submodule. The clock source bit CLKSRC in the CTRL1 Register defines whether the internal clock is connected to the output of a crystal oscillator (Oscillator Clock) or to the Peripheral Clock (generally from a PLL). In order to guarantee reliable operation, the clock source should be selected while the module is in Disable Mode (bit MDIS set in the Module Configuration Register).

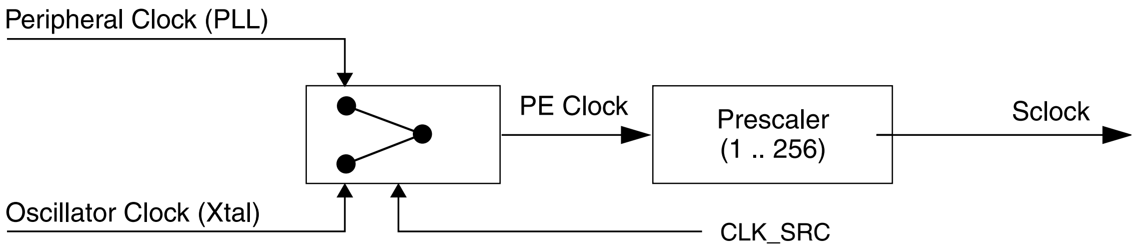


Figure 46-272. CAN engine clocking scheme

The crystal oscillator clock should be selected whenever a tight tolerance (up to 0.1%) is required in the CAN bus timing. The crystal oscillator clock has better jitter performance than PLL generated clocks.

The FlexCAN module supports a variety of means to setup bit timing parameters that are required by the CAN protocol. The Control Register has various fields used to control bit timing parameters: PRESDIV, PROPSEG, PSEG1, PSEG2 and RJW. See the description of the Control 1 Register (CTRL1).

The PRESDIV field controls a prescaler that generates the Serial Clock (Sclock), whose period defines the 'time quantum' used to compose the CAN waveform. A time quantum is the atomic unit of time handled by the CAN engine.

$$f_{Tq} = \frac{f_{CANCLK}}{(\text{Prescaler Value})}$$

A bit time is subdivided into three segments[2] (see Figure 46-273 and Table 46-292):

- SYNC_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section

2. For further explanation of the underlying concepts, see ISO/DIS 11519–1, Section 10.3. See also the CAN 2.0A/B protocol specification for bit timing.

- Time Segment 1: This segment includes the Propagation Segment and the Phase Segment 1 of the CAN standard. It can be programmed by setting the PROPSEG and the PSEG1 fields of the CTRL1 Register so that their sum (plus 2) is in the range of 4 to 16 time quanta
- Time Segment 2: This segment represents the Phase Segment 2 of the CAN standard. It can be programmed by setting the PSEG2 field of the CTRL1 Register (plus 1) to be 2 to 8 time quanta long

$$\text{Bit Rate} = \frac{f_{Tq}}{\text{(number of Time Quanta)}}$$

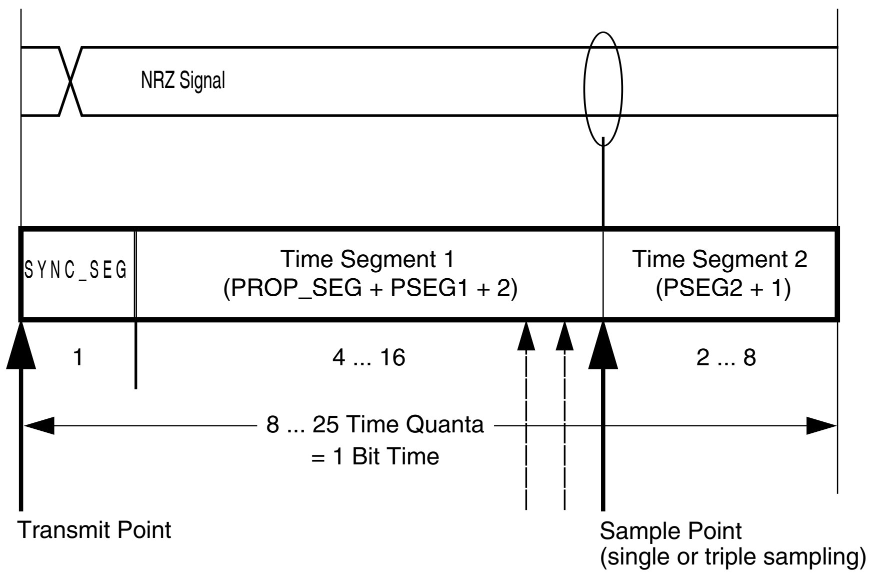


Figure 46-273. Segments within the bit time

Whenever CAN bit is used as a measure of duration (e.g. MCR[FRZACK] and MCR[LPMACK]), the number of peripheral clocks in one CAN bit can be calculated as:

$$NCCP = \frac{f_{SYS} \times [1 + (PSEG1 + 1) + (PSEG2 + 1) + (PROPSEG + 1)] \times (PRESDIV + 1)}{f_{CANCLK}}$$

where:

- NCCP is the number of peripheral clocks in one CAN bit;
- f_{CANCLK} is the Protocol Engine (PE) Clock (see Figure "CAN Engine Clocking Scheme"), in Hz;
- f_{SYS} is the frequency of operation of the system (CHI) clock, in Hz;
- PSEG1 is the value in CTRL1[PSEG1] field;
- PSEG2 is the value in CTRL1[PSEG2] field;

- PROPSEG is the value in CTRL1[PROPSEG] field;
- PRESDIV is the value in CTRL1[PRESDIV] field.

For example, 180 CAN bits = 180 x NCCP peripheral clock periods.

Table 46-292. Time segment syntax

Syntax	Description
SYNC_SEG	System expects transitions to occur on the bus during this period.
Transmit Point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample Point	A node samples the bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

The following table gives an overview of the CAN compliant segment settings and the related parameter values.

Table 46-293. CAN standard compliant bit time segment settings

Time segment 1	Time segment 2	Re-synchronization jump width
5 .. 10	2	1 .. 2
4 .. 11	3	1 .. 3
5 .. 12	4	1 .. 4
6 .. 13	5	1 .. 4
7 .. 14	6	1 .. 4
8 .. 15	7	1 .. 4
9 .. 16	8	1 .. 4

Note

The user must ensure the bit time settings are in compliance with the CAN standard. For bit time calculations, use an IPT (Information Processing Time) of 2, which is the value implemented in the FlexCAN module.

46.4.8.5 Arbitration and matching timing

During normal reception and transmission of frames, the matching, arbitration, move-in and move-out processes are executed during certain time windows inside the CAN frame, as shown in the following figures.

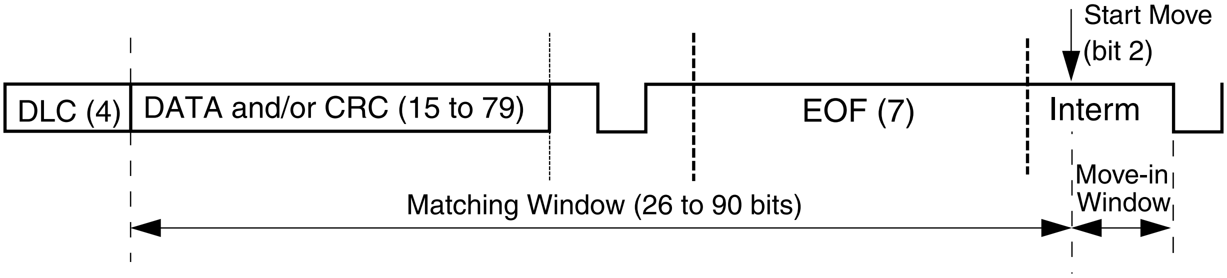


Figure 46-274. Matching and move-in time windows

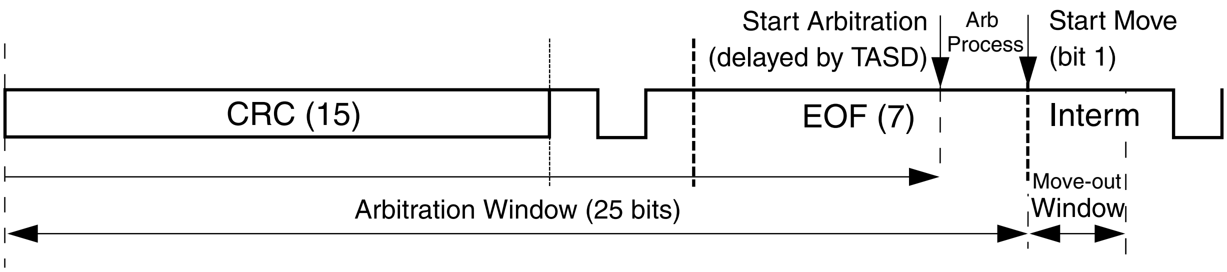


Figure 46-275. Arbitration and move-out time windows

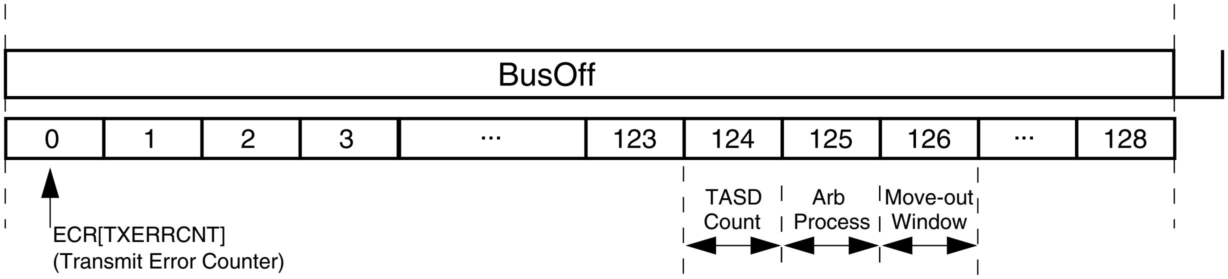


Figure 46-276. Arbitration at the end of bus off and move-out time windows

NOTE

The matching and arbitration timing shown in the preceding figures do not take into account the delay caused by the concurrent memory access due to the CPU or other internal FlexCAN sub-blocks.

When doing matching and arbitration, FlexCAN needs to scan the whole Message Buffer memory during the available time slot. In order to have sufficient time to do that, the following requirements must be observed:

- A valid CAN bit timing must be programmed, as indicated in Table 46-293
- The peripheral clock frequency can not be smaller than the oscillator clock frequency, that is, the PLL can not be programmed to divide down the oscillator clock; see Clock domains and restrictions
- There must be a minimum ratio between the peripheral clock frequency and the CAN bit rate, as specified in the following table:

Table 46-294. Minimum ratio between peripheral clock frequency and CAN bit rate

Number of Message Buffers	RFEN	Minimum number of peripheral clocks per CAN bit
16 and 32	0	16
64	0	25
16	1	16
32	1	17
64	1	30

A direct consequence of the first requirement is that the minimum number of time quanta per CAN bit must be 8, therefore the oscillator clock frequency should be at least 8 times the CAN bit rate. The minimum frequency ratio specified in the preceding table can be achieved by choosing a high enough peripheral clock frequency when compared to the oscillator clock frequency, or by adjusting one or more of the bit timing parameters (PRESDIV, PROPSEG, PSEG1, PSEG2) contained in the Control 1 Register (CTRL1).

In case of synchronous operation (when the peripheral clock frequency is equal to the oscillator clock frequency), the number of peripheral clocks per CAN bit can be adjusted by selecting an adequate value for PRESDIV in order to meet the requirement in the preceding table. In case of asynchronous operation (the peripheral clock frequency greater than the oscillator clock frequency), the number of peripheral clocks per CAN bit can be adjusted by both PRESDIV and/or the frequency ratio.

As an example, taking the case of 64 MBs, if the oscillator and peripheral clock frequencies are equal and the CAN bit timing is programmed to have 8 time quanta per bit, then the prescaler factor (PRESDIV + 1) should be at least 2. For prescaler factor equal to one and CAN bit timing with 8 time quanta per bit, the ratio between peripheral and oscillator clock frequencies should be at least 2.

46.4.9 Clock domains and restrictions

The FlexCAN module has two clock domains asynchronous to each other:

- The Bus Domain feeds the Control Host Interface (CHI) submodule and is derived from the peripheral clock.
- The Oscillator Domain feeds the CAN Protocol Engine (PE) submodule and is derived directly from a crystal oscillator clock, so that very low jitter performance can be achieved on the CAN bus.

When CTRL1[CLKSRC] bit is set, synchronous operation occurs because both domains are connected to the peripheral clock (creating a 1:1 ratio between the peripheral and oscillator domain clocks).

When the two domains are connected to clocks with different frequencies and/or phases, there are restrictions on the frequency relationship between the two clock domains. In the case of asynchronous operation, the Bus Domain clock frequency must always be greater than the Oscillator Domain clock frequency.

NOTE

Asynchronous operation with a 1:1 ratio between peripheral and oscillator clocks is not allowed.

46.4.10 Modes of operation details

The FlexCAN module has functional modes and low-power modes. See Modes of operation for an introductory description of all the modes of operation. The following sub-sections contain functional details on Freeze mode and the low-power modes.

CAUTION

“Permanent Dominant” failure on CAN Bus line is not supported by FlexCAN. If a Low-Power request or Freeze mode request is done during a “Permanent Dominant”, the corresponding acknowledge can never be asserted.

46.4.10.1 Freeze mode

This mode is requested by the CPU through the assertion of the HALT bit in the MCR Register or when the MCU is put into Debug mode. In both cases it is also necessary that the FRZ bit is asserted in the MCR Register and the module is not in a low-power mode. The acknowledgement is obtained through the assertion by the FlexCAN of FRZ_ACK bit in the same register. The CPU must only consider the FlexCAN in Freeze mode when both request and acknowledgement conditions are satisfied.

When Freeze mode is requested during transmission or reception, FlexCAN does the following:

- Waits to be in either Intermission, Passive Error, Bus Off or Idle state
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.

- Ignores the Rx input pin and drives the Tx pin as recessive
- Stops the prescaler, thus halting all CAN protocol activities
- Grants write access to the Error Counters Register, which is read-only in other modes
- Sets the NOT_RDY and FRZ_ACK bits in MCR

After requesting Freeze mode, the user must wait for the FRZ_ACK bit to be asserted in MCR before executing any other action, otherwise FlexCAN may operate in an unpredictable way. In Freeze mode, all memory mapped registers are accessible, except for CTRL1[CLK_SRC] bit that can be read but cannot be written.

Exiting Freeze mode is done in one of the following ways:

- CPU negates the FRZ bit in the MCR Register
- The MCU is removed from Debug Mode and/or the HALT bit is negated

The FRZ_ACK bit is negated after the protocol engine recognizes the negation of the freeze request. When out of Freeze mode, FlexCAN tries to re-synchronize to the CAN bus by waiting for 11 consecutive recessive bits.

46.4.10.2 Module Disable mode

This low power mode is normally used to temporarily disable a complete FlexCAN block, with no power consumption. It is requested by the CPU through the assertion of the MDIS bit in the MCR Register and the acknowledgement is obtained through the assertion by the FlexCAN of the LPM_ACK bit in the same register. The CPU must only consider the FlexCAN in Disable mode when both request and acknowledgement conditions are satisfied.

If the module is disabled during Freeze mode, it requests to disable the clocks to the PE and CHI sub-modules, sets the LPM_ACK bit and negates the FRZ_ACK bit.

If the module is disabled during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and then checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive

- Shuts down the clocks to the PE and CHI sub-modules
- Sets the NOTRDY and LPMACK bits in MCR

The Bus Interface Unit continues to operate, enabling the CPU to access memory mapped registers, except the Rx Mailboxes Global Mask Registers, the Rx Buffer 14 Mask Register, the Rx Buffer 15 Mask Register, the Rx FIFO Global Mask Register. The Rx FIFO Information Register, the Message Buffers, the Rx Individual Mask Registers, and the reserved words within RAM may not be accessed when the module is in Disable Mode. Exiting from this mode is done by negating the MDIS bit by the CPU, which causes the FlexCAN to request to resume the clocks and negate the LPM_ACK bit after the CAN protocol engine recognizes the negation of disable mode requested by the CPU.

46.4.10.3 Stop mode

This is a system low-power mode in which all MCU clocks can be stopped for maximum power savings. The Stop mode is globally requested by the CPU and the acknowledgement is obtained through the assertion by the FlexCAN of a Stop Acknowledgement signal. The CPU must only consider the FlexCAN in Stop mode when both request and acknowledgement conditions are satisfied.

If FlexCAN receives the global Stop mode request during Freeze mode, it sets the LPMACK bit, negates the FRZACK bit and then sends the Stop Acknowledge signal to the CPU, in order to shut down the clocks globally.

If Stop mode is requested during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive
- Sets the NOTRDY and LPMACK bits in MCR
- Sends a Stop Acknowledge signal to the CPU, so that it can shut down the clocks globally

Stop mode is exited when the CPU resumes the clocks and removes the Stop Mode request.

After the CAN protocol engine recognizes the negation of the Stop mode request, the FlexCAN negates the LPMACK bit. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up.

46.4.11 Interrupts

The module has many interrupt sources: interrupts due to message buffers and interrupts due to the ORed interrupts from MBs, Bus Off, Error, Tx Warning, and Rx Warning.

Each one of the message buffers can be an interrupt source, if its corresponding IMASK bit is set. There is no distinction between Tx and Rx interrupts for a particular buffer, under the assumption that the buffer is initialized for either transmission or reception. Each of the buffers has assigned a flag bit in the IFLAG Registers. The bit is set when the corresponding buffer completes a successful transfer and is cleared when the CPU writes it to 1 (unless another interrupt is generated at the same time).

Note

> It must be guaranteed that the CPU clears only the bit causing the current interrupt. For this reason, bit manipulation instructions (BSET) must not be used to clear interrupt flags. These instructions may cause accidental clearing of interrupt flags which are set after entering the current interrupt service routine.

If the Rx FIFO is enabled (MCR[RFEN] = 1), the interrupts corresponding to MBs 0 to 7 have different meanings. Bit 7 of the IFLAG1 becomes the "FIFO Overflow" flag; bit 6 becomes the FIFO Warning flag, bit 5 becomes the "Frames Available in FIFO flag" and bits 4-0 are unused. See the description of the Interrupt Flags 1 Register (IFLAG1) for more information.

For a combined interrupt where multiple MB interrupt sources are OR'd together, the interrupt is generated when any of the associated MBs (or FIFO, if applicable) generates an interrupt. In this case, the CPU must read the IFLAG registers to determine which MB or FIFO source caused the interrupt.

The interrupt sources for Bus Off, Error, Tx Warning and Rx Warning generate interrupts like the MB interrupt sources, and can be read from both the Error and Status Register 1 and 2. The Bus Off, Error, Tx Warning, and Rx Warning interrupt mask bits are located in the Control 1 Register.

46.4.12 Bus interface

The CPU access to FlexCAN registers are subject to the following rules:

- Unrestricted read and write access to supervisor registers (registers identified with S/U in Table "Module Memory Map" in Supervisor Mode or with S only) results in access error.
- Read and write access to implemented reserved address space results in access error.
- Write access to positions whose bits are all currently read-only results in access error. If at least one of the bits is not read-only then no access error is issued. Write permission to positions or some of their bits can change depending on the mode of operation or transitory state. Refer to register and bit descriptions for details.
- Read and write access to unimplemented address space results in access error.
- Read and write access to RAM located positions during Low Power Mode results in access error.
- If MAXMB is programmed with a value smaller than the available number of MBs, then the unused memory space can be used as general purpose RAM space. Note that reserved words within RAM cannot be used. As an example, suppose FlexCAN is configured with 16 MBs, RFFN is 0x0, and MAXMB is programmed with zero. The maximum number of MBs in this case becomes one. The RAM starts at 0x0080, and the space from 0x0080 to 0x008F is used by the one MB. The memory space from 0x0090 to 0x017F is available. The space between 0x0180 and 0x087F is reserved. The space from 0x0880 to 0x0883 is used by the one Individual Mask and the available memory in the Mask Registers space would be from 0x0884 to 0x08BF. From 0x08C0 through 0x09DF there are reserved words for internal use which cannot be used as general purpose RAM. As a general rule, free memory space for general purpose depends only on MAXMB.

46.4.13 Detection and Correction of Memory Errors

FlexCAN supports detection and correction of errors in memory read accesses. Each byte of FlexCAN memory is associated to 5 parity bits and the error correction mechanism assures that in this 13-bit word, errors in one bit can be corrected (corrected errors) and errors in 2 bits can be detected but not corrected (non-corrected errors). Errors in more than 2 bits may not be detected. When read, the parity bits are used to calculate a syndrome, which indicates the error in each byte.

Memory errors are indicated to Host through status register (Error Status Register (ERRSR)) and bus transfer errors, and reported through report registers (Error Report Address Register (RERRAR), Error Report Data Register (RERRDR) and Error Report Syndrome Register (RERRSYNR)).

The error detection and correction mechanism can be activated or not, controlled by ECCDIS bit in Memory Error Control Register (MECR). When disabled, updates on indications and reporting registers are stopped, but the parity bits are still calculated and written along with data in memory write operations. It assures that memory has consistent parity bits associated to data.

NOTE

All FlexCAN memory must be initialized before starting its operation in order to have the parity bits in memory properly updated. The WRMFRZ bit in Control 2 Register (CTRL2) grants write access to all memory positions from 0x080 to 0xADF.

To avoid that critical error correction configuration be accidentally changed, this protocol must be followed to enable the update of the Memory Error Control Register (MECR):

1. By default, ECRWRE bit in Control 2 Register (CTRL2) is 0 and ECRWRDIS bit in Memory Error Control Register (MECR) is 1.
2. Set ECRWRE bit in Control 2 Register (CTRL2).
3. Clear ECRWRDIS bit.
4. All writes to Memory Error Control Register (MECR) must keep ECRWRDIS cleared.
5. After configuration is done, lock the Memory Error Control Register (MECR) by either setting ECRWRDIS or clearing ECRWRE.

46.4.13.1 Sources of the Memory Access

The FlexCAN memory can be accessed by 2 major sources (or requestors):

- by Host (CPU): largest word accessed is 32-bit wide
- by FlexCAN internals processes (Match, Arbitration, RxMove on reception, TxMove on transmission): largest word accessed is 64-bit wide

The way that non-correctable errors are indicated and reported depends on the source of access.

46.4.13.2 Error Indication

Memory errors are indicated by flags HANCEIF, FANCEIF, CEIF in the Error Status Register (ERRSR). Non-correctable errors detected in memory reads requested by Host are indicated separatedly than the ones detected in requests by FlexCAN internal processes. Corrected errors indication makes no distinction of the source of the access. There are 3 independent flags for these 3 cases, and each flag will raise an interrupt unless it is masked by mask bits in Memory Error Control Register (MECR). If both non-corrected and corrected errors are found in different bytes in the same read operation both flags are set.

The non-corrected error detected in Host accesses are also indicated as a bus transfer error. A bus wait request may be asserted to extend the memory transaction to the moment the report registers are updated. This indication cannot be masked. If the flag bit HANCEIF is not masked, the same non-corrected error will raise a bus transfer error and an interrupt request.

Each indication flag has one Overrun flag in Error Status Register (ERRSR). The Overrun flags do not request interrupts. Overrun flags for non-correctable errors indicate that other errors of the same nature were detected after current error being treated, while overrun flags for correctable errors indicate that other errors of the same nature were detected before the current error being treated. This is the recommended handling sequence for error indication:

1. Get error report information from report registers
2. Use this information to take proper measures in the application
3. Clear the Interrupt flag
4. If the Overrun flag is active:
   a. Alert application that at least one error could not be handled
   b. Clear the Overrun flag

The FlexCAN internal processes can access memory in transactions larger than 32-bits. For the indication, this kind of access is considered a consecutive sequence of 32-bit accesses. If errors are found in 2 or more 32-bit words the Interrupt and Overrun flags are set simultaneously.

46.4.13.3 Error Reporting

Report registers Error Report Address Register (RERRAR), Error Report Data Register (RERRDR) and Error Report Syndrome Register (RERRSYNR) provide detailed information about the address read, raw data and syndrome read with error and indicated by the flags described in the “Error Indication” section. The address, data and syndrome registers are updated simultaneously along with the error flags, according to these rules.

1. If any of the 2 non-corrected error flags is currently set, does not update the registers (preserve the old non-corrected error reporting)
2. Else (no error flag is currently set or only the corrected error flag is currently set), update the report registers according to the new error; or according to the most severe of new errors if non-corrected and corrected errors are simultaneously detected

Reporting of errors detected in accesses larger than 32-bit follows the semantic described in the “Error Indication” section and in the Error Report Address Register (RERRAR) description.

The address reported in RERRAR and defined in ERRIAR are not the same listed in the module memory map. How the reported addresses correspond to locations in the module memory map is shown in the Error Injection Address Register (ERRIAR) description.

Addresses reported when reading memory portions organized as FIFOs, such as the Rx FIFO Structure and the Rx FIFO Information Register (RXFIR), refer to the address of the element accessed in the FIFO, not to the FIFO base address.

To assure coherence of the error report registers it is necessary to turn off the report update by clearing the MECR[RERRDIS] bit before reading the report registers.

46.4.13.4 Response to Errors

Correctable errors have no consequence on FlexCAN operation, once the corrected data is used by Host or FlexCAN internal processes. For host-initiated reads, a non-corrected error detected does not affect FlexCAN operation.

Non-corrected errors detected on memory reads requested by FlexCAN do not stop the FlexCAN processing and this may result in incorrect operation.

If a non-corrected error occurs during a reception process (Match or RxMove), an incorrect destination may be selected to store the incoming frame, a corrupted frame may be stored in the correct destination, or both. If a non-corrected error occurs during either an Arbitration or a TxMove process, either a non-highest priority frame may be mistakenly selected to be transmitted or no frame may be transmitted at all, when MECR[NCEFAFRZ] = 1.

The error report registers can provide information to the application for a customized handling of these situations. If the delay of this handling is not accepted and MECR[NCEFAFRZ] = 1, the FlexCAN stops operation automatically and enters Freeze mode when these errors are detected. See the Memory Error Control Register (MECR) description for more details.

46.4.13.5 Error Injection

The error injection registers ERRIAR, ERRIDPR, and ERRIPPR are used to inject errors in memory reads in order to force errors and consequently update the indication and reporting registers. How the error injection addresses correspond to locations in the module memory map is shown in the ERRIAR description.

The injection is done by flipping the data and parity bits correspondent to the bits in 1 in ERRIDPR and ERRIPPR. Injection can be selected specifically for memory accesses requested by Host or by FlexCAN internal processes.

In case of accesses larger than 32-bits, the EXTERRIE bit in Memory Error Control Register (MECR) extends the injection pattern, replicating it in 32-bit words to fill the width of the access.

NOTE

It is very unlikely, but error injection may correct a bit with error. This will not raise the error flags and reports as expected.

To ensure coherence among error injection registers and avoid spurious error injections, the HAERRIE and FAERRIE bits in Memory Error Control Register (MECR) must be cleared while configuring the memory injection registers.

46.5 Initialization/application information

This section provide instructions for initializing the FlexCAN module.

46.5.1 FlexCAN initialization sequence

The FlexCAN module may be reset as follows:

- MCU level hard reset, which resets all memory mapped registers asynchronously
- SOFTRST bit in MCR, which resets some of the memory mapped registers synchronously. See Table 46-2 to see what registers are affected by soft reset.
- MCU level soft reset, which has the same effect as the SOFTRST bit in MCR

Soft reset is synchronous and has to follow an internal request/acknowledge procedure across clock domains. Therefore, it may take some time to fully propagate its effects. The SOFTRST bit remains asserted while soft reset is pending, so software can poll this bit to

know when the reset has completed. Also, soft reset can not be applied while clocks are shut down in a low power mode. The low power mode should be exited and the clocks resumed before applying soft reset.

The clock source (CLKSRC bit) should be selected while the module is in Disable mode. After the clock source is selected and the module is enabled (MDIS bit negated), FlexCAN automatically goes to Freeze mode. In Freeze mode, FlexCAN is un-synchronized to the CAN bus, the HALT and FRZ bits in MCR Register are set, the internal state machines are disabled and the FRZACK and NOTRDY bits in the MCR Register are set. The Tx pin is in recessive state and FlexCAN does not initiate any transmission or reception of CAN frames. Note that the Message Buffers and the Rx Individual Mask Registers are not affected by reset, so they are not automatically initialized.

For any configuration change/initialization it is required that FlexCAN is put into Freeze mode; see Freeze mode. The following is a generic initialization sequence applicable to the FlexCAN module:

- Initialize the Module Configuration Register
    - Enable the individual filtering per MB and reception queue features by setting the IRMQ bit
    - Enable the warning interrupts by setting the WRNEN bit
    - If required, disable frame self reception by setting the SRXDIS bit
    - Enable the Rx FIFO by setting the RFEN bit
    - Enable the abort mechanism by setting the AEN bit
    - Enable the local priority feature by setting the LPRIOEN bit
- Initialize the Control Register
    - Determine the bit timing parameters: PROPSEG, PSEG1, PSEG2, RJW
    - Determine the bit rate by programming the PRESDIV field
    - Determine the internal arbitration mode (LBUF bit)
- Initialize the Message Buffers
    - The Control and Status word of all Message Buffers must be initialized
    - If Rx FIFO was enabled, the ID filter table must be initialized
    - Other entries in each Message Buffer should be initialized as required

- Initialize the Rx Individual Mask Registers
- When ECC is enabled, follow the initialization note in Detection and Correction of Memory Errors
- Set required interrupt mask bits in the IMASK Registers (for all MB interrupts) and in CTRL Register (for Bus Off and Error interrupts)
- Negate the HALT bit in MCR

Starting with the last event, FlexCAN attempts to synchronize to the CAN bus.

Chapter 47
Serial Peripheral Interface (SPI)

47.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The serial peripheral interface (SPI) module provides a synchronous serial bus for communication between an MCU and an external peripheral device.

47.1.1 Block Diagram

The block diagram of this module is as follows:

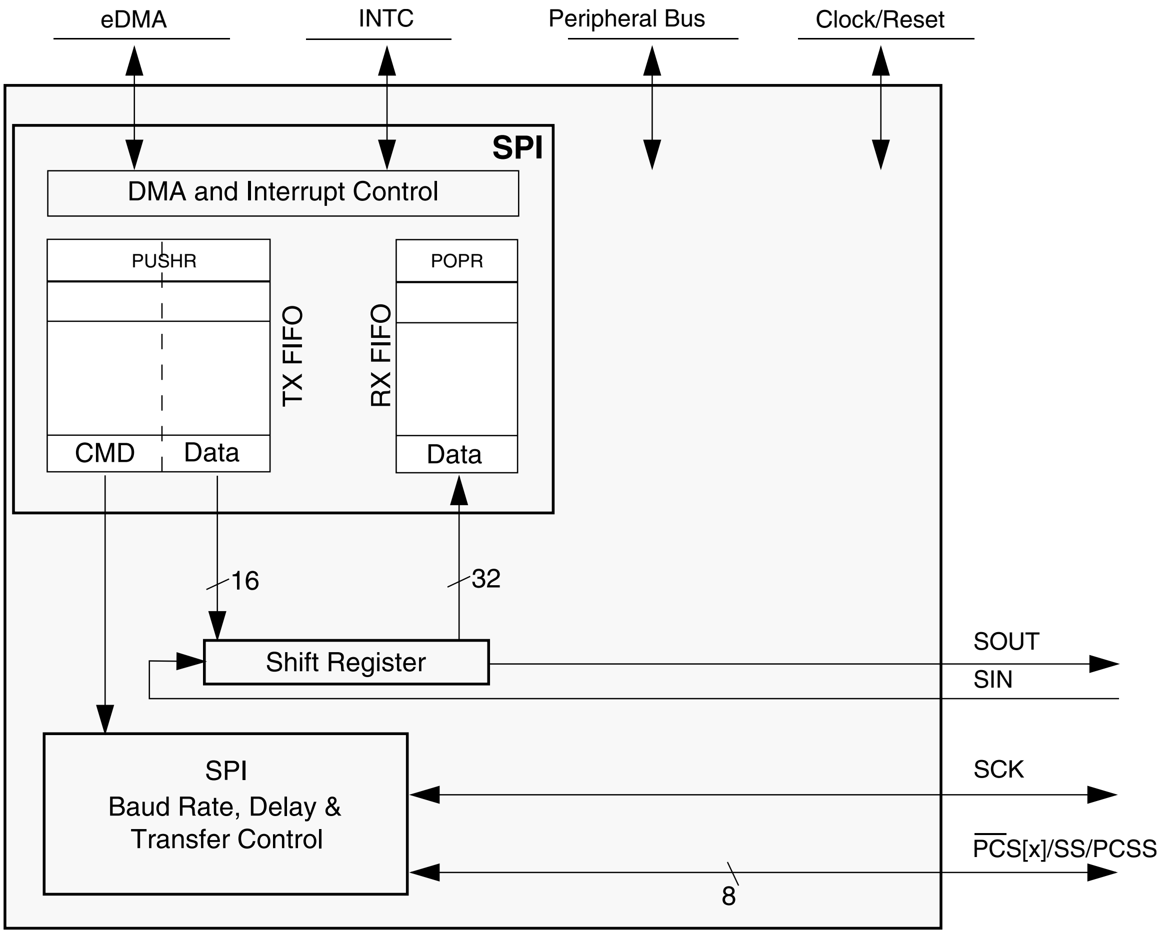


Figure 47-1. SPI Block Diagram

47.1.2 Features

The module supports the following features:

- Full-duplex, three-wire synchronous transfers
- Master and Slave modes
  - Data streaming operation in Slave mode with continuous slave selection
- Buffered transmit operation using the transmit first in first out (TX FIFO) with depth of four entries
- Buffered receive operation using the receive FIFO (RX FIFO) with depth of four entries
- TX and RX FIFOs can be disabled individually for low-latency updates to SPI queues

- Visibility into TX and RX FIFOs for ease of debugging
- Programmable transfer attributes on a per-frame basis:
  - four transfer attribute registers
  - Serial clock (SCK) with programmable polarity and phase
  - Various programmable delays
  - Programmable serial frame size of 4–16 bits, expandable by software control
    - SPI frames longer than 16 bits can be supported using the continuous selection format
  - Continuously held chip select capability
  - Parity control
- 6 peripheral chip selects (PCSs), expandable to 64 with external demultiplexer
- Deglitching support for up to 32 PCS with external demultiplexer
- DMA support for adding entries to TX FIFO and removing entries from RX FIFO:
  - TX FIFO is not full (TFFF)
  - RX FIFO is not empty (RFDF)
- Interrupt conditions:
  - End of Queue reached (EOQF)
  - TX FIFO is not full (TFFF)
  - Transfer of current frame complete (TCF)
  - Attempt to transmit with an empty Transmit FIFO (TFUF)
  - RX FIFO is not empty (RFDF)
  - Frame received while Receive FIFO is full (RFOF)
  - SPI Parity Error (SPEF)
- Global interrupt request line
- Modified SPI transfer formats for communication with slower peripheral devices
- Power-saving architectural features:
  - Support for Stop mode

47.1.3 SPI Configuration

The SPI configuration allows the module to send and receive serial data. This configuration allows the module to operate as a basic SPI block with internal FIFOs supporting external queue operation. Transmitted data and received data reside in separate FIFOs. The host CPU or a DMA controller read the received data from the Receive FIFO and write transmit data to the Transmit FIFO.

For queued operations, the SPI queues can reside in system RAM, external to the module. Data transfers between the queues and the module FIFOs are accomplished by a DMA controller or host CPU. The following figure shows a system example with DMA, SPI, and external queues in system RAM.

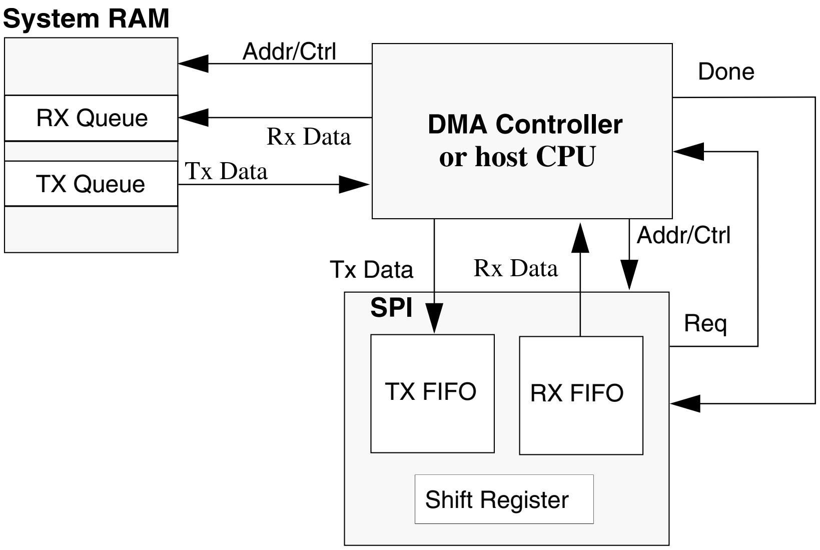


Figure 47-2. SPI with queues and DMA

47.1.4 Modes of Operation

The module supports the following modes of operation that can be divided into two categories:

- Module-specific modes:
  - Master mode
  - Slave mode
  - Module Disable mode
- MCU-specific modes:

- External Stop mode
- Debug mode

The module enters module-specific modes when the host writes a module register. The MCU-specific modes are controlled by signals external to the module. The MCU-specific modes are modes that an MCU may enter in parallel to the block-specific modes.

47.1.4.1 Master Mode

Master mode allows the module to initiate and control serial communication. In this mode, the SCK signal, SOUT signal, and the PCS[*x*] signals are controlled by the module and configured as outputs.

47.1.4.2 Slave Mode

Slave mode allows the module to communicate with SPI bus masters. In this mode, the module responds to externally controlled serial transfers. The SCK signal and the PCS[0]/$\overline{SS}$ signals are configured as inputs and driven by an SPI bus master.

47.1.4.3 Module Disable Mode

The Module Disable mode can be used for MCU power management. The clock to the non-memory mapped logic in the module can be stopped while in the Module Disable mode.

47.1.4.4 External Stop Mode

External Stop mode is used for MCU power management. The module supports the Peripheral Bus Stop mode mechanism. When a request is made to enter External Stop mode, it acknowledges the request and completes the transfer that is in progress. When the module reaches the frame boundary, it signals that the system clock to the module may be shut off.

47.1.4.5 Debug Mode

Debug mode is used for system development and debugging. The MCR[FRZ] bit controls module behavior in the Debug mode:

- If the bit is set, the module stops all serial transfers, when the MCU is in debug mode.
- If the bit is cleared, the MCU debug mode has no effect on the module.

47.2 Module signal descriptions

This section provides description of the module signals.

The following table lists the signals that may connect off chip depending on device implementation.

Table 47-1. Module signal descriptions

Signal	Master Mode	Slave Mode	Port Direction
PCS0/$\overline{\text{SS}}$	Peripheral Chip Select 0 output	Slave Select input	I/O
PCS[3:1]	Peripheral Chip Select 1 – 3	Unused	O
PCS4	Peripheral Chip Select 4	Unused	O
PCS5/ $\overline{\text{PCSS}}$	Peripheral Chip Select 5 /Peripheral Chip Select Strobe	Unused	O
SIN	Serial Data In	Serial Data In	I
SOUT	Serial Data Out	Serial Data Out	O
SCK	Master mode: Serial Clock (output)	Serial Clock (input)	I/O

47.2.1 PCS0/$\overline{\text{SS}}$ — Peripheral Chip Select/Slave Select

In Master mode, the PCS0 signal is an output that selects which slave device the current transmission is intended for.

In Slave mode, the active low $\overline{\text{SS}}$ signal is an input signal that allows an SPI master to select the module as the target for transmission.

47.2.2 PCS1 – PCS3 — Peripheral Chip Selects 1 – 3

PCS1 – PCS3 are output signals in Master mode.

In Slave mode, these signals are unused.

47.2.3 PCS4 — Peripheral Chip Select 4

In Master mode, PCS4 is an output signal.

In Slave mode, this signal is unused.

47.2.4 PCS5/$\overline{\text{PCSS}}$ — Peripheral Chip Select 5/Peripheral Chip Select Strobe

PCS5 is a Peripheral Chip Select output signal. When the module is in Master mode and the MCR[PCSSE] bit is cleared, this signal selects the slave device the current transfer is intended for.

When the module is in Master mode and the MCR[PCSSE] bit is set, the $\overline{\text{PCSS}}$ signal acts as a strobe to an external peripheral chip select demultiplexer, which decodes the PCS0 – PCS4 signals, preventing glitches on the demultiplexer outputs.

This signal is not used in Slave mode.

47.2.5 SIN — Serial Input

SIN is a serial data input signal.

NOTE

Serial Data Out output buffers are controller through SIUL, and cannot be controller through the module

47.2.6 SOUT — Serial Output

SOUT is a serial data output signal.

NOTE

Serial Data Out output buffers are controller through SIUL, and cannot be controller through the module.

47.2.7 SCK — Serial Clock

SCK is a serial communication clock signal.

In Master mode, the module generates the SCK.

In Slave mode, SCK is an input from an external bus master.

47.3 Memory Map/Register Definition

Register accesses to memory addresses that are reserved or undefined result in a transfer error. Write access to the POPR also results in a transfer error.

SPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_C000	Module Configuration Register (SPI0_MCR)	32	R/W	0000_4001h	47.3.1/2534
4002_C008	Transfer Count Register (SPI0_TCR)	32	R/W	0000_0000h	47.3.2/2537
4002_C00C	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR0)	32	R/W	7800_0000h	47.3.3/2537
4002_C00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI0_CTAR0_SLAVE)	32	R/W	7800_0000h	47.3.4/2542
4002_C010	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR1)	32	R/W	7800_0000h	47.3.3/2537
4002_C014	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR2)	32	R/W	7800_0000h	47.3.3/2537
4002_C018	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR3)	32	R/W	7800_0000h	47.3.3/2537
4002_C02C	Status Register (SPI0_SR)	32	R/W	0200_0000h	47.3.5/2544
4002_C030	DMA/Interrupt Request Select and Enable Register (SPI0_RSER)	32	R/W	0000_0000h	47.3.6/2547
4002_C034	PUSH TX FIFO Register In Master Mode (SPI0_PUSHR)	32	R/W	0000_0000h	47.3.7/2549
4002_C034	PUSH TX FIFO Register In Slave Mode (SPI0_PUSHR_SLAVE)	32	R/W	0000_0000h	47.3.8/2551
4002_C038	POP RX FIFO Register (SPI0_POPR)	32	R	0000_0000h	47.3.9/2552
4002_C03C	Transmit FIFO Registers (SPI0_TXFR0)	32	R	0000_0000h	47.3.10/ 2552
4002_C040	Transmit FIFO Registers (SPI0_TXFR1)	32	R	0000_0000h	47.3.10/ 2552
4002_C044	Transmit FIFO Registers (SPI0_TXFR2)	32	R	0000_0000h	47.3.10/ 2552
4002_C048	Transmit FIFO Registers (SPI0_TXFR3)	32	R	0000_0000h	47.3.10/ 2552
4002_C07C	Receive FIFO Registers (SPI0_RXFR0)	32	R	0000_0000h	47.3.11/ 2553
4002_C080	Receive FIFO Registers (SPI0_RXFR1)	32	R	0000_0000h	47.3.11/ 2553
4002_C084	Receive FIFO Registers (SPI0_RXFR2)	32	R	0000_0000h	47.3.11/ 2553
4002_C088	Receive FIFO Registers (SPI0_RXFR3)	32	R	0000_0000h	47.3.11/ 2553

SPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_D000	Module Configuration Register (SPI1_MCR)	32	R/W	0000_4001h	47.3.1/2534
4002_D008	Transfer Count Register (SPI1_TCR)	32	R/W	0000_0000h	47.3.2/2537
4002_D00C	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR0)	32	R/W	7800_0000h	47.3.3/2537
4002_D00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI1_CTAR0_SLAVE)	32	R/W	7800_0000h	47.3.4/2542
4002_D010	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR1)	32	R/W	7800_0000h	47.3.3/2537
4002_D014	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR2)	32	R/W	7800_0000h	47.3.3/2537
4002_D018	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR3)	32	R/W	7800_0000h	47.3.3/2537
4002_D02C	Status Register (SPI1_SR)	32	R/W	0200_0000h	47.3.5/2544
4002_D030	DMA/Interrupt Request Select and Enable Register (SPI1_RSER)	32	R/W	0000_0000h	47.3.6/2547
4002_D034	PUSH TX FIFO Register In Master Mode (SPI1_PUSHR)	32	R/W	0000_0000h	47.3.7/2549
4002_D034	PUSH TX FIFO Register In Slave Mode (SPI1_PUSHR_SLAVE)	32	R/W	0000_0000h	47.3.8/2551
4002_D038	POP RX FIFO Register (SPI1_POPR)	32	R	0000_0000h	47.3.9/2552
4002_D03C	Transmit FIFO Registers (SPI1_TXFR0)	32	R	0000_0000h	47.3.10/ 2552
4002_D040	Transmit FIFO Registers (SPI1_TXFR1)	32	R	0000_0000h	47.3.10/ 2552
4002_D044	Transmit FIFO Registers (SPI1_TXFR2)	32	R	0000_0000h	47.3.10/ 2552
4002_D048	Transmit FIFO Registers (SPI1_TXFR3)	32	R	0000_0000h	47.3.10/ 2552
4002_D07C	Receive FIFO Registers (SPI1_RXFR0)	32	R	0000_0000h	47.3.11/ 2553
4002_D080	Receive FIFO Registers (SPI1_RXFR1)	32	R	0000_0000h	47.3.11/ 2553
4002_D084	Receive FIFO Registers (SPI1_RXFR2)	32	R	0000_0000h	47.3.11/ 2553
4002_D088	Receive FIFO Registers (SPI1_RXFR3)	32	R	0000_0000h	47.3.11/ 2553
400A_C000	Module Configuration Register (SPI2_MCR)	32	R/W	0000_4001h	47.3.1/2534
400A_C008	Transfer Count Register (SPI2_TCR)	32	R/W	0000_0000h	47.3.2/2537
400A_C00C	Clock and Transfer Attributes Register (In Master Mode) (SPI2_CTAR0)	32	R/W	7800_0000h	47.3.3/2537
400A_C00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI2_CTAR0_SLAVE)	32	R/W	7800_0000h	47.3.4/2542
400A_C010	Clock and Transfer Attributes Register (In Master Mode) (SPI2_CTAR1)	32	R/W	7800_0000h	47.3.3/2537

Table continues on the next page...

SPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_C014	Clock and Transfer Attributes Register (In Master Mode) (SPI2_CTAR2)	32	R/W	7800_0000h	47.3.3/2537
400A_C018	Clock and Transfer Attributes Register (In Master Mode) (SPI2_CTAR3)	32	R/W	7800_0000h	47.3.3/2537
400A_C02C	Status Register (SPI2_SR)	32	R/W	0200_0000h	47.3.5/2544
400A_C030	DMA/Interrupt Request Select and Enable Register (SPI2_RSER)	32	R/W	0000_0000h	47.3.6/2547
400A_C034	PUSH TX FIFO Register In Master Mode (SPI2_PUSHR)	32	R/W	0000_0000h	47.3.7/2549
400A_C034	PUSH TX FIFO Register In Slave Mode (SPI2_PUSHR_SLAVE)	32	R/W	0000_0000h	47.3.8/2551
400A_C038	POP RX FIFO Register (SPI2_POPR)	32	R	0000_0000h	47.3.9/2552
400A_C03C	Transmit FIFO Registers (SPI2_TXFR0)	32	R	0000_0000h	47.3.10/ 2552
400A_C040	Transmit FIFO Registers (SPI2_TXFR1)	32	R	0000_0000h	47.3.10/ 2552
400A_C044	Transmit FIFO Registers (SPI2_TXFR2)	32	R	0000_0000h	47.3.10/ 2552
400A_C048	Transmit FIFO Registers (SPI2_TXFR3)	32	R	0000_0000h	47.3.10/ 2552
400A_C07C	Receive FIFO Registers (SPI2_RXFR0)	32	R	0000_0000h	47.3.11/ 2553
400A_C080	Receive FIFO Registers (SPI2_RXFR1)	32	R	0000_0000h	47.3.11/ 2553
400A_C084	Receive FIFO Registers (SPI2_RXFR2)	32	R	0000_0000h	47.3.11/ 2553
400A_C088	Receive FIFO Registers (SPI2_RXFR3)	32	R	0000_0000h	47.3.11/ 2553
400A_D000	Module Configuration Register (SPI3_MCR)	32	R/W	0000_4001h	47.3.1/2534
400A_D008	Transfer Count Register (SPI3_TCR)	32	R/W	0000_0000h	47.3.2/2537
400A_D00C	Clock and Transfer Attributes Register (In Master Mode) (SPI3_CTAR0)	32	R/W	7800_0000h	47.3.3/2537
400A_D00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI3_CTAR0_SLAVE)	32	R/W	7800_0000h	47.3.4/2542
400A_D010	Clock and Transfer Attributes Register (In Master Mode) (SPI3_CTAR1)	32	R/W	7800_0000h	47.3.3/2537
400A_D014	Clock and Transfer Attributes Register (In Master Mode) (SPI3_CTAR2)	32	R/W	7800_0000h	47.3.3/2537
400A_D018	Clock and Transfer Attributes Register (In Master Mode) (SPI3_CTAR3)	32	R/W	7800_0000h	47.3.3/2537
400A_D02C	Status Register (SPI3_SR)	32	R/W	0200_0000h	47.3.5/2544
400A_D030	DMA/Interrupt Request Select and Enable Register (SPI3_RSER)	32	R/W	0000_0000h	47.3.6/2547
400A_D034	PUSH TX FIFO Register In Master Mode (SPI3_PUSHR)	32	R/W	0000_0000h	47.3.7/2549

Table continues on the next page...

SPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_D034	PUSH TX FIFO Register In Slave Mode (SPI3_PUSHR_SLAVE)	32	R/W	0000_0000h	47.3.8/2551
400A_D038	POP RX FIFO Register (SPI3_POPR)	32	R	0000_0000h	47.3.9/2552
400A_D03C	Transmit FIFO Registers (SPI3_TXFR0)	32	R	0000_0000h	47.3.10/ 2552
400A_D040	Transmit FIFO Registers (SPI3_TXFR1)	32	R	0000_0000h	47.3.10/ 2552
400A_D044	Transmit FIFO Registers (SPI3_TXFR2)	32	R	0000_0000h	47.3.10/ 2552
400A_D048	Transmit FIFO Registers (SPI3_TXFR3)	32	R	0000_0000h	47.3.10/ 2552
400A_D07C	Receive FIFO Registers (SPI3_RXFR0)	32	R	0000_0000h	47.3.11/ 2553
400A_D080	Receive FIFO Registers (SPI3_RXFR1)	32	R	0000_0000h	47.3.11/ 2553
400A_D084	Receive FIFO Registers (SPI3_RXFR2)	32	R	0000_0000h	47.3.11/ 2553
400A_D088	Receive FIFO Registers (SPI3_RXFR3)	32	R	0000_0000h	47.3.11/ 2553

47.3.1 Module Configuration Register (SPIx_MCR)

Contains bits to configure various attributes associated with the module operations. The HALT and MDIS bits can be changed at any time, but the effect takes place only on the next frame boundary. Only the HALT and MDIS bits in the MCR can be changed, while the module is in the Running state.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSTR	CONT_SCKE	DCONF		FRZ	MTFE	PCSSE	ROOE	0		PCSIS					
W	MSTR	CONT_SCKE			FRZ	MTFE	PCSSE	ROOE			PCSIS					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MDIS	DIS_TXF	DIS_RXF	0	0	SMPL_PT		0					FCPCS	PES	HALT
W		MDIS	DIS_TXF	DIS_RXF	CLR_TXF	CLR_RXF	SMPL_PT							FCPCS	PES	HALT
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SPIx_MCR field descriptions

Field	Description
31 MSTR	Master/Slave Mode Select Configures the module for either Master mode or Slave mode. 0 The module is in Slave mode. 1 The module is in Master mode.
30 CONT_SCKE	Continuous SCK Enable Enables the Serial Communication Clock (SCK) to run continuously.

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
	0 Continuous SCK disabled. 1 Continuous SCK enabled.
29–28 DCONF	SPI Configuration. Selects among the different configurations of the module. 00 SPI 01 Reserved 10 Reserved 11 Reserved
27 FRZ	Freeze Enables the SPI transfers to be stopped on the next frame boundary when the device enters Debug mode. 0 Do not halt serial transfers in Debug mode. 1 Halt serial transfers in Debug mode.
26 MTFE	Modified Timing Format Enable Enables a modified transfer format to be used. 0 Modified SPI transfer format disabled. 1 Modified SPI transfer format enabled.
25 PCSSE	Peripheral Chip Select Strobe Enable Enables the PCS5/ $\overline{\text{PCSS}}$ to operate as a PCS Strobe output signal. 0 PCS5/ $\overline{\text{PCSS}}$ is used as the Peripheral Chip Select[5] signal. 1 PCS5/ $\overline{\text{PCSS}}$ is used as an active-low PCS Strobe signal.
24 ROOE	Receive FIFO Overflow Overwrite Enable In the RX FIFO overflow condition, configures the module to ignore the incoming serial data or overwrite existing data. If the RX FIFO is full and new data is received, the data from the transfer, generating the overflow, is ignored or shifted into the shift register. 0 Incoming data is ignored. 1 Incoming data is shifted into the shift register.
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 PCSIS	Peripheral Chip Select x Inactive State Determines the inactive state of PCSx. 0 The inactive state of PCSx is low. 1 The inactive state of PCSx is high.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 MDIS	Module Disable Allows the clock to be stopped to the non-memory mapped logic in the module effectively putting it in a software-controlled power-saving state. The reset value of the MDIS bit is parameterized, with a default

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
	reset value of 0. When SPI is used in Slave Mode, it is recommended to leave this bit set to '0', since a slave doesn't have control over master transactions. 0 Enables the module clocks. 1 Allows external logic to disable the module clocks.
13 DIS_TXF	Disable Transmit FIFO When the TX FIFO is disabled, the transmit part of the module operates as a simplified double-buffered SPI. This bit can be written only when the MDIS bit is cleared. 0 TX FIFO is enabled. 1 TX FIFO is disabled.
12 DIS_RXF	Disable Receive FIFO When the RX FIFO is disabled, the receive part of the module operates as a simplified double-buffered SPI. This bit can only be written when the MDIS bit is cleared. 0 RX FIFO is enabled. 1 RX FIFO is disabled.
11 CLR_TXF	Clear TX FIFO Flushes the TX FIFO. Writing a 1 to CLR_TXF clears the TX FIFO Counter. The CLR_TXF bit is always read as zero. 0 Do not clear the TX FIFO counter. 1 Clear the TX FIFO counter.
10 CLR_RXF	Flushes the RX FIFO. Writing a 1 to CLR_RXF clears the RX Counter. The CLR_RXF bit is always read as zero. 0 Do not clear the RX FIFO counter. 1 Clear the RX FIFO counter.
9–8 SMPL_PT	Sample Point Controls when the module master samples SIN in Modified Transfer Format. This field is valid only when CPHA bit in CTARn[CPHA] is 0. 00 0 system clocks between SCK edge and SIN sample 01 1 system clock between SCK edge and SIN sample 10 2 system clocks between SCK edge and SIN sample 11 Reserved
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 FCPCS	Fast Continuous PCS Mode. This bit enables the masking of “After SCK (t_{ASC})” and “PCS to SCK (t_{CSC})” delays when operating in Continuous PCS mode. This masking is not available if Continuous SCK mode is enabled. The individual delay masks are selected via bits MASC and MCSC of the PUSHR register. The firmware should select appropriate masks when providing continuous frames via the PUSHR register. 0 Normal or Slow Continuous PCS mode. Masking of delays is disabled. 1 Fast Continuous PCS mode. Delays masked via control bits in PUSHR register.

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
1 PES	Parity Error Stop Controls SPI operation when a parity error is detected in a received SPI frame. 0 SPI frame transmission continues. 1 SPI frame transmission stops.
0 HALT	Halt The HALT bit starts and stops SPI transfers. See Start and Stop of Module transfers 0 Start transfers. 1 Stop transfers.

47.3.2 Transfer Count Register (SPIx_TCR)

TCR contains a counter that indicates the number of SPI transfers made. The transfer counter is intended to assist in queue management. Do not write the TCR when the module is in the Running state.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPI_TCNT																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_TCR field descriptions

Field	Description
31–16 SPI_TCNT	SPI Transfer Counter Counts the number of SPI transfers the module makes. The SPI_TCNT field increments every time the last bit of an SPI frame is transmitted. A value written to SPI_TCNT presets the counter to that value. SPI_TCNT is reset to zero at the beginning of the frame when the CTCNT field is set in the executing SPI command. The Transfer Counter wraps around; incrementing the counter past 65535 resets the counter to zero.
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

47.3.3 Clock and Transfer Attributes Register (In Master Mode) (SPIx_CTAR*n*)

CTAR registers are used to define different transfer attributes. Do not write to the CTAR registers while the module is in the Running state.

In Master mode, the CTAR registers define combinations of transfer attributes such as frame size, clock phase and polarity, data bit ordering, baud rate, and various delays. In slave mode, a subset of the bitfields in CTAR0 are used to set the slave transfer attributes.

When the module is configured as an SPI master, the CTAS field in the command portion of the TX FIFO entry selects which of the CTAR registers is used. When the module is configured as an SPI bus slave, the CTAR0 register is used.

Address: Base address + Ch offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	DBR	FMSZ				CPOL	CPHA	LSBFE	PCSSCK		PASC		PDT		PBR	
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	CSSCK				ASC				DT				BR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_CTAR*n* field descriptions

Field	Description
31 DBR	Double Baud Rate Doubles the effective baud rate of the Serial Communications Clock (SCK). This field is used only in master mode. It effectively halves the Baud Rate division ratio, supporting faster frequencies, and odd division ratios for the Serial Communications Clock (SCK). When the DBR bit is set, the duty cycle of the Serial Communications Clock (SCK) depends on the value in the Baud Rate Prescaler and the Clock Phase bit as listed in the following table. See the BR field description for details on how to compute the baud rate.

Table 47-48. SPI SCK Duty Cycle

DBR	CPHA	PBR	SCK Duty Cycle
0	any	any	50/50
1	0	00	50/50
1	0	01	33/66
1	0	10	40/60
1	0	11	43/57
1	1	00	50/50
1	1	01	66/33
1	1	10	60/40
1	1	11	57/43

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

Field	Description
	0 The baud rate is computed normally with a 50/50 duty cycle. 1 The baud rate is doubled with the duty cycle depending on the Baud Rate Prescaler.
30–27 FMSZ	Frame Size The number of bits transferred per frame is equal to the FMSZ field value plus 1. Regardless of the transmission mode, the minimum valid frame size value is 4.
26 CPOL	Clock Polarity Selects the inactive state of the Serial Communications Clock (SCK). This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock polarities. When the Continuous Selection Format is selected, switching between clock polarities without stopping the module can cause errors in the transfer due to the peripheral device interpreting the switch of clock polarity as a valid clock edge. 0 The inactive state value of SCK is low. 1 The inactive state value of SCK is high.
25 CPHA	Clock Phase Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode , the bit value is ignored and the transfers are done as if the CPHA bit is set to 1. 0 Data is captured on the leading edge of SCK and changed on the following edge. 1 Data is changed on the leading edge of SCK and captured on the following edge.
24 LSBFE	LSB First Specifies whether the LSB or MSB of the frame is transferred first. 0 Data is transferred MSB first. 1 Data is transferred LSB first.
23–22 PCSSCK	PCS to SCK Delay Prescaler Selects the prescaler value for the delay between assertion of PCS and the first edge of the SCK. See the CSSCK field description for information on how to compute the PCS to SCK Delay. Refer PCS to SCK Delay (t_{CSC}) for more details. 00 PCS to SCK Prescaler value is 1. 01 PCS to SCK Prescaler value is 3. 10 PCS to SCK Prescaler value is 5. 11 PCS to SCK Prescaler value is 7.
21–20 PASC	After SCK Delay Prescaler Selects the prescaler value for the delay between the last edge of SCK and the negation of PCS. See the ASC field description for information on how to compute the After SCK Delay. Refer After SCK Delay (t_{ASC}) for more details. 00 Delay after Transfer Prescaler value is 1. 01 Delay after Transfer Prescaler value is 3. 10 Delay after Transfer Prescaler value is 5. 11 Delay after Transfer Prescaler value is 7.

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

Field	Description
19–18 PDT	Delay after Transfer Prescaler Selects the prescaler value for the delay between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. The PDT field is only used in master mode. See the DT field description for details on how to compute the Delay after Transfer. Refer Delay after Transfer (t_{DT}) for more details. 00 Delay after Transfer Prescaler value is 1. 01 Delay after Transfer Prescaler value is 3. 10 Delay after Transfer Prescaler value is 5. 11 Delay after Transfer Prescaler value is 7.
17–16 PBR	Baud Rate Prescaler Selects the prescaler value for the baud rate. This field is used only in master mode. The baud rate is the frequency of the SCK. The system clock is divided by the prescaler value before the baud rate selection takes place. See the BR field description for details on how to compute the baud rate. 00 Baud Rate Prescaler value is 2. 01 Baud Rate Prescaler value is 3. 10 Baud Rate Prescaler value is 5. 11 Baud Rate Prescaler value is 7.
15–12 CSSCK	PCS to SCK Delay Scaler Selects the scaler value for the PCS to SCK delay. This field is used only in master mode. The PCS to SCK Delay is the delay between the assertion of PCS and the first edge of the SCK. The delay is a multiple of the system clock period, and it is computed according to the following equation: $t_{CSC} = (1/f_{SYS}) \times PCSSCK \times CSSCK$. The following table lists the delay scaler values.

Table 47-47. Delay Scaler Encoding

Field Value	Delay Scaler Value
0000	2
0001	4
0010	8
0011	16
0100	32
0101	64
0110	128
0111	256
1000	512
1001	1024
1010	2048
1011	4096
1100	8192
1101	16384

Table continues on the next page...

SPIx_CTARn field descriptions (continued)

Field	Description
	Table 47-47. Delay Scaler Encoding (continued) Field Value: 1110 — Delay Scaler Value: 32768 Field Value: 1111 — Delay Scaler Value: 65536 Refer PCS to SCK Delay (t_{CSC}) for more details.
11–8 ASC	After SCK Delay Scaler Selects the scaler value for the After SCK Delay. This field is used only in master mode. The After SCK Delay is the delay between the last edge of SCK and the negation of PCS. The delay is a multiple of the system clock period, and it is computed according to the following equation: $t_{ASC} = (1/f_{SYS}) \times PASC \times ASC$ See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values. Refer After SCK Delay (t_{ASC}) for more details.
7–4 DT	Delay After Transfer Scaler Selects the Delay after Transfer Scaler. This field is used only in master mode. The Delay after Transfer is the time between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. In the Continuous Serial Communications Clock operation, the DT value is fixed to one SCK clock period, The Delay after Transfer is a multiple of the system clock period, and it is computed according to the following equation: $t_{DT} = (1/f_{SYS}) \times PDT \times DT$ See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values.
3–0 BR	Baud Rate Scaler Selects the scaler value for the baud rate. This field is used only in master mode. The prescaled system clock is divided by the Baud Rate Scaler to generate the frequency of the SCK. The baud rate is computed according to the following equation: SCK baud rate = $(f_{SYS} /PBR) \times [(1+DBR)/BR]$ The following table lists the baud rate scaler values.

Table 47-46. Baud Rate Scaler

CTARn[BR]	Baud Rate Scaler Value
0000	2
0001	4
0010	6
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

Field	Description
	Table 47-46. Baud Rate Scaler (continued)

CTARn[BR]	Baud Rate Scaler Value
1001	512
1010	1024
1011	2048
1100	4096
1101	8192
1110	16384
1111	32768

47.3.4 Clock and Transfer Attributes Register (In Slave Mode) (SPIx_CTAR*n*_SLAVE)

When the module is configured as an SPI bus slave, the CTAR0 register is used.

Address: Base address + Ch offset + (0d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FMSZ					CPOL	CPHA	PE	PP	0	0					
W																
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_CTAR*n*_SLAVE field descriptions

Field	Description
31–27 FMSZ	Frame Size The number of bits transfered per frame is equal to the FMSZ field value plus 1. Note that the minimum valid value of frame size is 4.
26 CPOL	Clock Polarity

Table continues on the next page...

SPIx_CTAR*n*_SLAVE field descriptions (continued)

Field	Description
	Selects the inactive state of the Serial Communications Clock (SCK). 0 The inactive state value of SCK is low. 1 The inactive state value of SCK is high.
25 CPHA	Clock Phase Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode , the bit value is ignored and the transfers are done as the CPHA bit is set to 1. 0 Data is captured on the leading edge of SCK and changed on the following edge. 1 Data is changed on the leading edge of SCK and captured on the following edge.
24 PE	Parity Enable Enables parity bit transmission and reception for the frame. 0 No parity bit included/checked. 1 Parity bit is transmitted instead of last data bit in frame, parity checked for received frame.
23 PP	Parity Polarity Controls polarity of the parity bit transmitted and checked. 0 Even Parity: the number of 1 bits in the transmitted frame is even. The SR[SPEF] bit is set if the number of 1 bits is odd in the received frame. 1 Odd Parity: the number of 1 bits in the transmitted frame is odd. The SR[SPEF] bit is set if the number of 1 bits is even in the received frame.
22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

47.3.5 Status Register (SPIx_SR)

SR contains status and flag bits. The bits reflect the status of the module and indicate the occurrence of events that can generate interrupt or DMA requests. Software can clear flag bits in the SR by writing a 1 to them. Writing a 0 to a flag bit has no effect. This register may not be writable in Module Disable mode due to the use of power saving mechanisms.

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCF	TXRXS	0	EOQF	TFUF	0	TFFF	0	0	0	SPEF	0	RFOF	0	RFDF	0
W	w1c	w1c		w1c	w1c		w1c				w1c		w1c			
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCTR				TXNXTPTR				RXCTR				POPNXTPTR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_SR field descriptions

Field	Description
31 TCF	Transfer Complete Flag Indicates that all bits in a frame have been shifted out. TCF remains set until it is cleared by writing a 1 to it. 0 Transfer not complete. 1 Transfer complete.
30 TXRXS	TX and RX Status Reflects the run status of the module.

Table continues on the next page...

SPIx_SR field descriptions (continued)

Field	Description
	0 Transmit and receive operations are disabled (The module is in Stopped state). 1 Transmit and receive operations are enabled (The module is in Running state).
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 EOQF	End of Queue Flag Indicates that the last entry in a queue has been transmitted when the module is in Master mode. The EOQF bit is set when the TX FIFO entry has the EOQ bit set in the command halfword and the end of the transfer is reached. The EOQF bit remains set until cleared by writing a 1 to it. When the EOQF bit is set, the TXRXS bit is automatically cleared. 0 EOQ is not set in the executing command. 1 EOQ is set in the executing SPI command.
27 TFUF	Transmit FIFO Underflow Flag Indicates an underflow condition in the TX FIFO. The transmit underflow condition is detected only for SPI blocks operating in Slave mode and SPI configuration. TFUF is set when the TX FIFO of the module operating in SPI Slave mode is empty and an external SPI master initiates a transfer. The TFUF bit remains set until cleared by writing 1 to it. 0 No TX FIFO underflow. 1 TX FIFO underflow has occurred.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 TFFF	Transmit FIFO Fill Flag Provides a method for the module to request more entries to be added to the TX FIFO. The TFFF bit is set while the TX FIFO is not full. The TFFF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller to the TX FIFO full request. 0 TX FIFO is full. 1 TX FIFO is not full.
24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 SPEF	SPI Parity Error Flag Indicates that a SPI frame with parity error had been received. The bit remains set until it is cleared by writing a 1 to it. 0 No parity error. 1 Parity error has occurred.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 RFOF	Receive FIFO Overflow Flag Indicates an overflow condition in the RX FIFO. The field is set when the RX FIFO and shift register are full and a transfer is initiated. The bit remains set until it is cleared by writing a 1 to it.

Table continues on the next page...

SPIx_SR field descriptions (continued)

Field	Description
	0 No Rx FIFO overflow. 1 Rx FIFO overflow has occurred.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 RFDF	Receive FIFO Drain Flag Provides a method for the module to request that entries be removed from the RX FIFO. The bit is set while the RX FIFO is not empty. The RFDF bit can be cleared by acknowledgement from the DMA controller when the RX FIFO is empty. 0 RX FIFO is empty. 1 This bit auto-clears on every RXFR read performed.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 TXCTR	TX FIFO Counter Indicates the number of valid entries in the TX FIFO. The TXCTR is incremented every time the PUSHR is written. The TXCTR is decremented every time an SPI command is executed and the SPI data is transferred to the shift register.
11–8 TXNXTPTR	Transmit Next Pointer Indicates which TX FIFO entry is transmitted during the next transfer. The TXNXTPTR field is updated every time SPI data is transferred from the TX FIFO to the shift register.
7–4 RXCTR	RX FIFO Counter Indicates the number of entries in the RX FIFO. The RXCTR is decremented every time the POPR is read. The RXCTR is incremented every time data is transferred from the shift register to the RX FIFO.
3–0 POPNXTPTR	Pop Next Pointer Contains a pointer to the RX FIFO entry to be returned when the POPR is read. The POPNXTPTR is updated when the POPR is read.

47.3.6 DMA/Interrupt Request Select and Enable Register (SPIx_RSER)

RSER controls DMA and interrupt requests. Do not write to the RSER while the module is in the Running state.

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCF_RE	0	0	EOQF_RE	TFUF_RE	0	TFFF_RE	TFFF_DIRS	0	0	SPEF_RE	0	RFOF_RE	0	RFDF_RE	RFDF_DIRS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_RSER field descriptions

Field	Description
31 TCF_RE	Transmission Complete Request Enable Enables TCF flag in the SR to generate an interrupt request. 0 TCF interrupt requests are disabled. 1 TCF interrupt requests are enabled.
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 EOQF_RE	Finished Request Enable Enables the EOQF flag in the SR to generate an interrupt request. 0 EOQF interrupt requests are disabled. 1 EOQF interrupt requests are enabled.
27 TFUF_RE	Transmit FIFO Underflow Request Enable Enables the TFUF flag in the SR to generate an interrupt request. 0 TFUF interrupt requests are disabled. 1 TFUF interrupt requests are enabled.

Table continues on the next page...

SPIx_RSER field descriptions (continued)

Field	Description
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 TFFF_RE	Transmit FIFO Fill Request Enable Enables the TFFF flag in the SR to generate a request. The TFFF_DIRS bit selects between generating an interrupt request or a DMA request. 0 TFFF interrupts or DMA requests are disabled. 1 TFFF interrupts or DMA requests are enabled.
24 TFFF_DIRS	Transmit FIFO Fill DMA or Interrupt Request Select Selects between generating a DMA request or an interrupt request. When SR[TFFF] and RSER[TFFF_RE] are set, this field selects between generating an interrupt request or a DMA request. 0 TFFF flag generates interrupt requests. 1 TFFF flag generates DMA requests.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 SPEF_RE	SPI Parity Error Request Enable Enables the SPEF flag in the SR to generate an interrupt request. 0 SPEF interrupt requests are disabled. 1 SPEF interrupt requests are enabled.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 RFOF_RE	Receive FIFO Overflow Request Enable Enables the RFOF flag in the SR to generate an interrupt request. 0 RFOF interrupt requests are disabled. 1 RFOF interrupt requests are enabled.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 RFDF_RE	Receive FIFO Drain Request Enable Enables the RFDF flag in the SR to generate a request. The RFDF_DIRS bit selects between generating an interrupt request or a DMA request. 0 RFDF interrupt or DMA requests are disabled. 1 RFDF interrupt or DMA requests are enabled.
16 RFDF_DIRS	Receive FIFO Drain DMA or Interrupt Request Select Selects between generating a DMA request or an interrupt request. When the RFDF flag bit in the SR is set, and the RFDF_RE bit in the RSER is set, the RFDF_DIRS bit selects between generating an interrupt request or a DMA request. 0 Interrupt request. 1 DMA request.

Table continues on the next page...

SPIx_RSER field descriptions (continued)

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

47.3.7 PUSH TX FIFO Register In Master Mode (SPIx_PUSHR)

PUSHR provides the means to write to the TX FIFO . Data written to this register is transferred to the TX FIFO 8- or 16-bit write accesses to the PUSHR transfer all 32 register bits to the TXFIFO.The register structure is different in Master and Slave modes. In Master mode, the register provides 16-bit command and data to the TX FIFO. In Slave mode, all 32 register bits can be used as data, supporting up to 32-bit SPI Frame operation.

A PUSHR Read Operation returns the topmost TX FIFO entry.

When the module is disabled, any writes to this register will not update the FIFO. Hence any reads performed during Module disable mode will return the last PUSHR write performed when Module was enabled.

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CONT	CTAS			EOQ	CTCNT	PE_MASC	PP_MCSC	0		PCS					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXDATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_PUSHR field descriptions

Field	Description
31 CONT	Continuous Peripheral Chip Select Enable

Table continues on the next page...

SPIx_PUSHR field descriptions (continued)

Field	Description
	Selects a continuous selection format. The bit is used in SPI Master mode. The bit enables the selected PCS signals to remain asserted between transfers. 0 Return PCSn signals to their inactive state between transfers. 1 Keep PCSn signals asserted between transfers.
30–28 CTAS	Clock and Transfer Attributes Select Selects which CTAR to use in master mode to specify the transfer attributes for the associated SPI frame. In SPI Slave mode, CTAR0 is used. See the chapter on chip configuration to determine how many CTARs this device has. You should not program a value in this field for a register that is not present. 000 CTAR0 001 CTAR1 010 CTAR2 011 CTAR3 100 Reserved 101 Reserved 110 Reserved 111 Reserved
27 EOQ	End Of Queue Host software uses this bit to signal to the module that the current SPI transfer is the last in a queue. At the end of the transfer, the EOQF bit in the SR is set. 0 The SPI data is not the last data to transfer. 1 The SPI data is the last data to transfer.
26 CTCNT	Clear Transfer Counter Clears the TCNT field in the TCR register. The TCNT field is cleared before the module starts transmitting the current SPI frame. 0 Do not clear the TCR[TCNT] field. 1 Clear the TCR[TCNT] field.
25 PE_MASC	Parity Enable or Mask T_{ASC} delay in the current frame PE – This bit enables parity bit transmission and parity reception check for the SPI frame. MASC - The current frame has the “after SCK” delay masked if this bit is asserted. See Fast Continuous Selection Format for more details. **NOTE:** This bit is used as Mask T_{ASC} in the Fast Continuous PCS mode when MCR[FCPCS] is set. 0 PE - No parity bit included/checked. MASC - T_{ASC} delay is not masked and the current frame has the after SCK delay. 1 PE - Parity bit is transmitted instead of the last data bit in the frame; parity is checked for the received frame. MASC - T_{ASC} delay is masked in the current frame.
24 PP_MCSC	Parity Polarity or Mask T_{CSC} delay in the next frame PP - It controls the polarity of the parity bit transmitted and checked. MCSC - The next frame has the “PCS to SCK” delay masked if this bit is asserted. See Fast Continuous Selection Format for more details.

Table continues on the next page...

SPIx_PUSHR field descriptions (continued)

Field	Description
	NOTE: This bit is used as Mask T_{CSC} in the Fast Continuous PCS mode when MCR[FCPCS] is set. 0 PP - Even Parity: the number of 1 bits in the transmitted frame is even. The SR[SPEF] bit is set if the number of 1 bits is odd in the received frame. MCSC - T_{CSC} delay is not masked and the next frame has the PCS to SCK delay. 1 PP - Odd Parity: the number of 1 bits in the transmitted frame is odd. The SR[SPEF] bit is set if the number of 1 bits is even in the received frame. MCSC - T_{CSC} delay is masked in the next frame.
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 PCS	Select which PCS signals are to be asserted for the transfer. Refer to the chip configuration chapter for the number of PCS signals used in this MCU. 0 Negate the PCS[x] signal. 1 Assert the PCS[x] signal.
15–0 TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command.

47.3.8 PUSH TX FIFO Register In Slave Mode (SPIx_PUSHR_SLAVE)

PUSHR provides the means to write to the TX FIFO. Data written to this register is transferred to the TX FIFO. 8- or 16-bit write accesses to the PUSHR transfer all 32 register bits to the TXFIFO.The register structure is different in master and slave modes. In master mode the register provides 16-bit command and data to the TX FIFO. In slave mode, all 32 register bits can be used as data, supporting up to 32-bit SPI Frame operation.

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_PUSHR_SLAVE field descriptions

Field	Description
31–0 TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command.

47.3.9 POP RX FIFO Register (SPIx_POPR)

POPR is used to read the RX FIFO. Eight- or sixteen-bit read accesses to the POPR have the same effect on the RX FIFO as 32-bit read accesses. A write to this register will generate a Transfer Error.

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_POPR field descriptions

Field	Description
31–0 RXDATA	Received Data Contains the SPI data from the RX FIFO entry to which the Pop Next Data Pointer points.

47.3.10 Transmit FIFO Registers (SPIx_TXFR*n*)

TXFRn registers provide visibility into the TX FIFO for debugging purposes. Each register is an entry in the TX FIFO. The registers are read-only and cannot be modified. Reading the TXFRx registers does not alter the state of the TX FIFO.

Address: Base address + 3Ch offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCMD_TXDATA																TXDATA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_TXFR*n* field descriptions

Field	Description
31–16 TXCMD_ TXDATA	Transmit Command or Transmit Data In Master mode the TXCMD field contains the command that sets the transfer attributes for the SPI data.In Slave mode, the TXDATA contains 16 MSB bits of the SPI data to be shifted out.
15–0 TXDATA	Transmit Data Contains the SPI data to be shifted out.

47.3.11 Receive FIFO Registers (SPIx_RXFR*n*)

RXFRn provide visibility into the RX FIFO for debugging purposes. Each register is an entry in the RX FIFO. The RXFR registers are read-only. Reading the RXFRx registers does not alter the state of the RX FIFO.

Address: Base address + 7Ch offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_RXFR*n* field descriptions

Field	Description
31–0 RXDATA	Receive Data Contains the received SPI data.

47.4 Functional description

The Serial Peripheral Interface (SPI) block supports full-duplex, synchronous serial communications between MCUs and peripheral devices. The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes.

The SPI has the following configurations

- The SPI Configuration in which the module operates as a basic SPI or a queued SPI.

The DCONF field in the Module Configuration Register (MCR) determines the module Configuration. SPI configuration is selected when DCONF within SPIx_MCR is 0b00.

The CTARn registers hold clock and transfer attributes. The SPI configuration allows to select which CTAR to use on a frame by frame basis by setting a field in the SPI command.

See Clock and Transfer Attributes Register (In Master Mode) (SPI_CTAR*n*) for information on the fields of CTAR registers.

Typical master to slave connections are shown in the following figure. When a data transfer operation is performed, data is serially shifted a predetermined number of bit positions. Because the modules are linked, data is exchanged between the master and the slave. The data that was in the master shift register is now in the shift register of the slave, and vice versa. At the end of a transfer, the Transfer Control Flag(TCF) bit in the Shift Register(SR) is set to indicate a completed frame transfer.

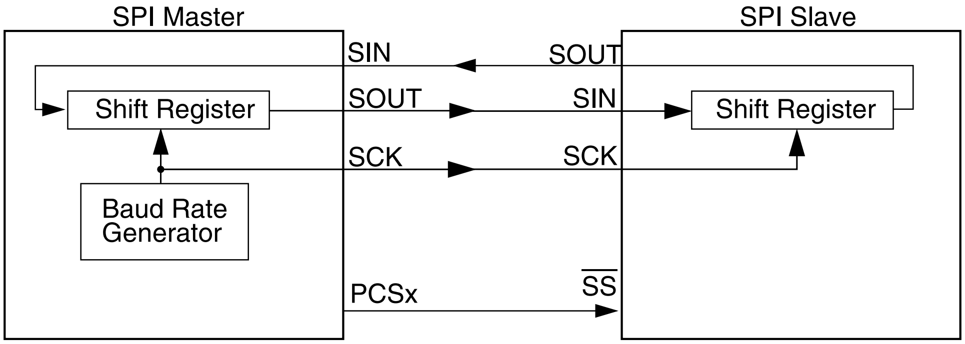


Figure 47-123. SPI serial protocol overview

Generally, more than one slave device can be connected to the module master. Six Peripheral Chip Select (PCS) signals of the module masters can be used to select which of the slaves to communicate with. Refer to the chip configuration chapter for the number of PCS signals used in this MCU.

The SPI configuration share transfer protocol and timing properties which are described independently of the configuration in Transfer formats . The transfer rate and delay settings are described in Module baud rate and clock delay generation.

47.4.1 Start and Stop of module transfers

The module has two operating states: Stopped and Running. Both the states are independent of it's configuration. The default state of the module is Stopped. In the Stopped state, no serial transfers are initiated in Master mode and no transfers are responded to in Slave mode. The Stopped state is also a safe state for writing the various configuration registers of the module without causing undetermined results. In the Running state serial transfers take place.

The TXRXS bit in the SR indicates the state of module. The bit is set if the module is in Running state.

The module starts or transitions to Running when all of the following conditions are true:

- SR[EOQF] bit is clear

- MCU is not in the Debug mode or the MCR[FRZ] bit is clear
- MCR[HALT] bit is clear

The module stops or transitions from Running to Stopped after the current frame when any one of the following conditions exist:

- SR[EOQF] bit is set
- MCU in the Debug mode and the MCR[FRZ] bit is set
- MCR[HALT] bit is set

State transitions from Running to Stopped occur on the next frame boundary if a transfer is in progress, or immediately if no transfers are in progress.

47.4.2 Serial Peripheral Interface (SPI) configuration

The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes. The module is in SPI configuration when the DCONF field in the MCR is 0b00. The SPI frames can be 32 bits long. The host CPU or a DMA controller transfers the SPI data from the external to SPI RAM queues to a TX FIFO buffer. The received data is stored in entries in the RX FIFO buffer. The host CPU or the DMA controller transfers the received data from the RX FIFO to memory external to the module. The operation of FIFO buffers is described in Transmit First In First Out (TX FIFO) buffering mechanism, Transmit First In First Out (TX FIFO) buffering mechanism and Receive First In First Out (RX FIFO) buffering mechanism. The interrupt and DMA request conditions are described in Interrupts/DMA requests.

The SPI configuration supports two block-specific modes—Master mode and Slave mode.In Master mode the module initiates and controls the transfer according to the fields of the executing SPI Command. In Slave mode, the module responds only to transfers initiated by a bus master external to it and the SPI command field space is reserved.

47.4.2.1 Master mode

In SPI Master, mode the module initiates the serial transfers by controlling the SCK and the PCS signals. The executing SPI Command determines which CTARs will be used to set the transfer attributes and which PCS signals to assert . The command field also contains various bits that help with queue management and transfer protocol . See PUSH TX FIFO Register In Master Mode (SPI_PUSHR) for details on the SPI command fields.

The data in the executing TX FIFO entry is loaded into the shift register and shifted out on the Serial Out (SOUT) pin. In SPI Master mode, each SPI frame to be transmitted has a command associated with it, allowing for transfer attribute control on a frame by frame basis.

47.4.2.2 Slave mode

In SPI Slave mode the module responds to transfers initiated by an SPI bus master. It does not initiate transfers. Certain transfer attributes such as clock polarity, clock phase, and frame size must be set for successful communication with an SPI master. The SPI Slave mode transfer attributes are set in the CTAR0 . The data is shifted out with MSB first. Shifting out of LSB is not supported in this mode.

47.4.2.3 FIFO disable operation

The FIFO disable mechanisms allow SPI transfers without using the TX FIFO or RX FIFO. The module operates as a double-buffered simplified SPI when the FIFOs are disabled. The Transmit and Receive side of the FIFOs are disabled separately; setting the MCR[DIS_TXF] bit disables the TX FIFO, and setting the MCR[DIS_RXF] bit disables the RX FIFO.

The FIFO disable mechanisms are transparent to the user and to host software. Transmit data and commands are written to the PUSHR and received data is read from the POPR.

When the TX FIFO is disabled, the fields SR[TFFF], SR[TFUF] and SR[TXCTR] behave as if there is a one-entry FIFO but the contents of TXFRs, SR[TXNXTPTR] are undefined. Similarly, when the RX FIFO is disabled, the RFDF, RFOF, and RXCTR fields in the SR behave as if there is a one-entry FIFO, but the contents of the RXFR registers and POPNXTPTR are undefined.

47.4.2.4 Transmit First In First Out (TX FIFO) buffering mechanism

The TX FIFO functions as a buffer of SPI data for transmission. The TX FIFO holds four words, each consisting of SPI data. The number of entries in the TX FIFO is device-specific. SPI data is added to the TX FIFO by writing to the Data Field of module PUSH FIFO Register (PUSHR). TX FIFO entries can only be removed from the TX FIFO by being shifted out or by flushing the TX FIFO.

The TX FIFO Counter field (TXCTR) in the module Status Register (SR) indicates the number of valid entries in the TX FIFO. The TXCTR is updated every time a 8- or 16-bit write takes place to the Data Field of SPI_PUSHR or SPI data is transferred into the shift register from the TX FIFO.

The TXNXTPTR field indicates the TX FIFO Entry that will be transmitted during the next transfer. The TXNXTPTR field is incremented every time SPI data is transferred from the TX FIFO to the shift register. The maximum value of the field is equal to the maximum implemented TXFR number and it rolls over after reaching the maximum.

47.4.2.4.1 Filling the TX FIFO

Host software or other intelligent blocks can add (push) entries to the TX FIFO by writing to the PUSHR. When the TX FIFO is not full, the TX FIFO Fill Flag (TFFF) in the SR is set. The TFFF bit is cleared when TX FIFO is full and the DMA controller indicates that a write to PUSHR is complete. Writing a '1' to the TFFF bit also clears it. The TFFF can generate a DMA request or an interrupt request. See Transmit FIFO Fill Interrupt or DMA Request for details.

The module ignores attempts to push data to a full TX FIFO, and the state of the TX FIFO does not change and no error condition is indicated.

47.4.2.4.2 Draining the TX FIFO

The TX FIFO entries are removed (drained) by shifting SPI data out through the shift register. Entries are transferred from the TX FIFO to the shift register and shifted out as long as there are valid entries in the TX FIFO. Every time an entry is transferred from the TX FIFO to the shift register, the TX FIFO Counter decrements by one. At the end of a transfer, the TCF bit in the SR is set to indicate the completion of a transfer. The TX FIFO is flushed by writing a '1' to the CLR_TXF bit in MCR.

If an external bus master initiates a transfer with a module slave while the slave's TX FIFO is empty, the Transmit FIFO Underflow Flag (TFUF) in the slave's SR is set. See Transmit FIFO Underflow Interrupt Request for details.

47.4.2.5 Receive First In First Out (RX FIFO) buffering mechanism

The RX FIFO functions as a buffer for data received on the SIN pin. The RX FIFO holds four received SPI data frames. The number of entries in the RX FIFO is device-specific. SPI data is added to the RX FIFO at the completion of a transfer when the received data in the shift register is transferred into the RX FIFO. SPI data are removed (popped) from

the RX FIFO by reading the module POP RX FIFO Register (POPR). RX FIFO entries can only be removed from the RX FIFO by reading the POPR or by flushing the RX FIFO.

The RX FIFO Counter field (RXCTR) in the module's Status Register (SR) indicates the number of valid entries in the RX FIFO. The RXCTR is updated every time the POPR is read or SPI data is copied from the shift register to the RX FIFO.

The POPNXTPTR field in the SR points to the RX FIFO entry that is returned when the POPR is read. The POPNXTPTR contains the positive offset from RXFR0 in a number of 32-bit registers. For example, POPNXTPTR equal to two means that the RXFR2 contains the received SPI data that will be returned when the POPR is read. The POPNXTPTR field is incremented every time the POPR is read. The maximum value of the field is equal to the maximum implemented RXFR number and it rolls over after reaching the maximum.

47.4.2.5.1 Filling the RX FIFO

The RX FIFO is filled with the received SPI data from the shift register. While the RX FIFO is not full, SPI frames from the shift register are transferred to the RX FIFO. Every time an SPI frame is transferred to the RX FIFO, the RX FIFO Counter is incremented by one.

If the RX FIFO and shift register are full and a transfer is initiated, the RFOF bit in the SR is set indicating an overflow condition. Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

47.4.2.5.2 Draining the RX FIFO

Host CPU or a DMA can remove (pop) entries from the RX FIFO by reading the module POP RX FIFO Register (POPR). A read of the POPR decrements the RX FIFO Counter by one. Attempts to pop data from an empty RX FIFO are ignored and the RX FIFO Counter remains unchanged. The data, read from the empty RX FIFO, is undetermined.

When the RX FIFO is not empty, the RX FIFO Drain Flag (RFDF) in the SR is set. The RFDF bit is cleared when the RX_FIFO is empty and the DMA controller indicates that a read from POPR is complete or by writing a 1 to it.

47.4.3 Module baud rate and clock delay generation

The SCK frequency and the delay values for serial transfer are generated by dividing the system clock frequency by a prescaler and a scaler with the option for doubling the baud rate. The following figure shows conceptually how the SCK signal is generated.

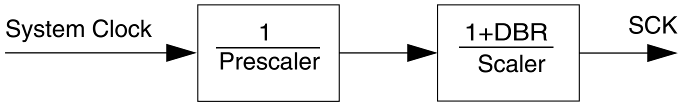


Figure 47-124. Communications clock prescalers and scalers

47.4.3.1 Baud rate generator

The baud rate is the frequency of the SCK. The system clock is divided by a prescaler (PBR) and scaler (BR) to produce SCK with the possibility of halving the scaler division. The DBR, PBR, and BR fields in the CTARs select the frequency of SCK by the formula in the BR field description. The following table shows an example of how to compute the baud rate.

Table 47-202. Baud rate computation example

f_{SYS}	PBR	Prescaler	BR	Scaler	DBR	Baud rate
100 MHz	0b00	2	0b0000	2	0	25 Mb/s
20 MHz	0b00	2	0b0000	2	1	10 Mb/s

NOTE

The clock frequencies mentioned in the preceding table are given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

47.4.3.2 PCS to SCK Delay (t_{CSC})

The PCS to SCK delay is the length of time from assertion of the PCS signal to the first SCK edge. See Figure 47-126 for an illustration of the PCS to SCK delay. The PCSSCK and CSSCK fields in the CTAR*x* registers select the PCS to SCK delay by the formula in the CSSCK field description. The following table shows an example of how to compute the PCS to SCK delay.

Table 47-203. PCS to SCK delay computation example

f_{SYS}	PCSSCK	Prescaler	CSSCK	Scaler	PCS to SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 μs

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

47.4.3.3 After SCK Delay (t_{ASC})

The After SCK Delay is the length of time between the last edge of SCK and the negation of PCS. See Figure 47-126 and Figure 47-127 for illustrations of the After SCK delay. The PASC and ASC fields in the CTAR*x* registers select the After SCK Delay by the formula in the ASC field description. The following table shows an example of how to compute the After SCK delay.

Table 47-204. After SCK Delay computation example

f_{SYS}	PASC	Prescaler	ASC	Scaler	After SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 μs

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

47.4.3.4 Delay after Transfer (t_{DT})

The Delay after Transfer is the minimum time between negation of the PCS signal for a frame and the assertion of the PCS signal for the next frame. See Figure 47-126 for an illustration of the Delay after Transfer. The PDT and DT fields in the CTAR*x* registers select the Delay after Transfer by the formula in the DT field description. The following table shows an example of how to compute the Delay after Transfer.

Table 47-205. Delay after Transfer computation example

f_{SYS}	PDT	Prescaler	DT	Scaler	Delay after Transfer
100 MHz	0b01	3	0b1110	32768	0.98 ms

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

When in Non-Continuous Clock mode the t_{DT} delay is configured according to the equation specified in the CTAR[DT] field description. When in Continuous Clock mode, the delay is fixed at 1 SCK period.

47.4.3.5 Peripheral Chip Select Strobe Enable ($\overline{\text{PCSS}}$)

The $\overline{\text{PCSS}}$ signal provides a delay to allow the PCS signals to settle after a transition occurs thereby avoiding glitches. When the Module is in Master mode and the PCSSE bit is set in the MCR, $\overline{\text{PCSS}}$ provides a signal for an external demultiplexer to decode the PCS[0] – PCS[4] signals into as many as 128 glitch-free PCS signals. The following figure shows the timing of the $\overline{\text{PCSS}}$ signal relative to PCS signals.

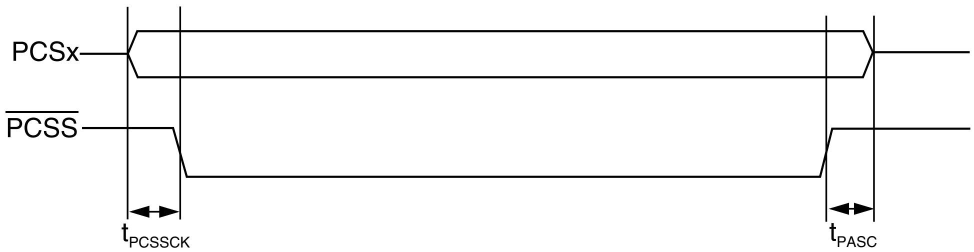


Figure 47-125. Peripheral Chip Select Strobe timing

The delay between the assertion of the PCS signals and the assertion of $\overline{\text{PCSS}}$ is selected by the PCSSCK field in the CTAR based on the following formula:

$$t_{\text{PCSSCK}} = \frac{1}{f_{\text{SYS}}} \times \text{PCSSCK}$$

At the end of the transfer, the delay between $\overline{\text{PCSS}}$ negation and PCS negation is selected by the PASC field in the CTAR based on the following formula:

$$t_{\text{PASC}} = \frac{1}{f_{\text{SYS}}} \times \text{PASC}$$

The following table shows an example of how to compute the t_{pcssck} delay.

Table 47-206. Peripheral Chip Select Strobe Assert computation example

f_{SYS}	PCSSCK	Prescaler	Delay before Transfer
100 MHz	0b11	7	70.0 ns

The following table shows an example of how to compute the t_{pasc} delay.

Table 47-207. Peripheral Chip Select Strobe Negate computation example

f_{SYS}	PASC	Prescaler	Delay after Transfer
100 MHz	0b11	7	70.0 ns

The $\overline{\text{PCSS}}$ signal is not supported when Continuous Serial Communication SCK mode is enabled.

NOTE

The clock frequency mentioned in the preceding tables is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

47.4.4 Transfer formats

The SPI serial communication is controlled by the Serial Communications Clock (SCK) signal and the PCS signals. The SCK signal provided by the master device synchronizes shifting and sampling of the data on the SIN and SOUT pins. The PCS signals serve as enable signals for the slave devices.

In Master mode, the CPOL and CPHA bits in the Clock and Transfer Attributes Registers (CTARn) select the polarity and phase of the serial clock, SCK.

- CPOL - Selects the idle state polarity of the SCK
- CPHA - Selects if the data on SOUT is valid before or on the first SCK edge

Even though the bus slave does not control the SCK signal, in Slave mode the values of CPOL and CPHA must be identical to the master device settings to ensure proper transmission. In SPI Slave mode, only CTAR0 is used.

The module supports four different transfer formats:

- Classic SPI with CPHA=0
- Classic SPI with CPHA=1
- Modified Transfer Format with CPHA = 0
- Modified Transfer Format with CPHA = 1

A modified transfer format is supported to allow for high-speed communication with peripherals that require longer setup times. The module can sample the incoming data later than halfway through the cycle to give the peripheral more setup time. The MTFE bit in the MCR selects between Classic SPI Format and Modified Transfer Format.

In the SPI configurations, the module provides the option of keeping the PCS signals asserted between frames. See Continuous Selection Format for details.

47.4.4.1 Classic SPI Transfer Format (CPHA = 0)

The transfer format shown in following figure is used to communicate with peripheral SPI slave devices where the first data bit is available on the first clock edge. In this format, the master and slave sample their SIN pins on the odd-numbered SCK edges and change the data on their SOUT pins on the even-numbered SCK edges.

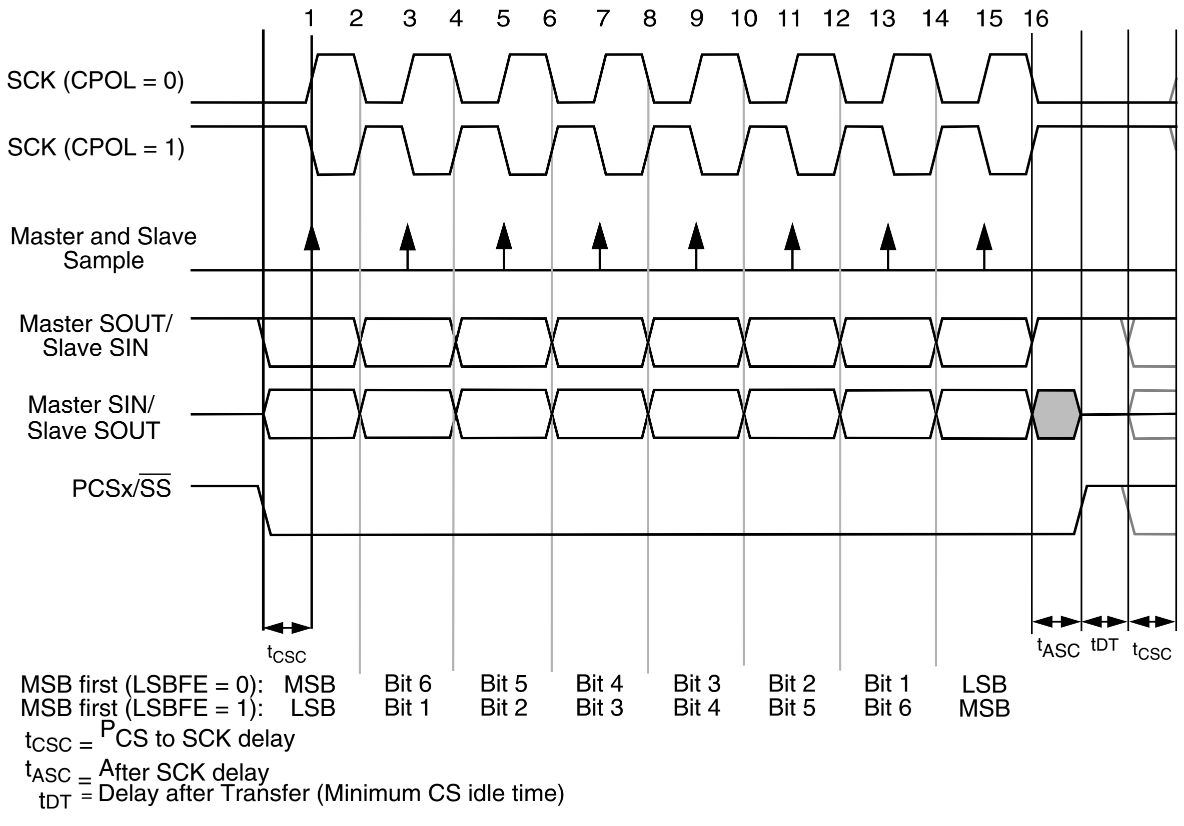


Figure 47-126. Module transfer timing diagram (MTFE=0, CPHA=0, FMSZ=8)

The master initiates the transfer by placing its first data bit on the SOUT pin and asserting the appropriate peripheral chip select signals to the slave device. The slave responds by placing its first data bit on its SOUT pin. After the t_{ASC} delay elapses, the master outputs the first edge of SCK. The master and slave devices use this edge to sample the first input data bit on their serial data input signals. At the second edge of the SCK, the master and

slave devices place their second data bit on their serial data output signals. For the rest of the frame the master and the slave sample their SIN pins on the odd-numbered clock edges and changes the data on their SOUT pins on the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signals. A delay of t_{DT} is inserted before a new frame transfer can be initiated by the master.

47.4.4.2 Classic SPI Transfer Format (CPHA = 1)

This transfer format shown in the following figure is used to communicate with peripheral SPI slave devices that require the first SCK edge before the first data bit becomes available on the slave SOUT pin. In this format, the master and slave devices change the data on their SOUT pins on the odd-numbered SCK edges and sample the data on their SIN pins on the even-numbered SCK edges .

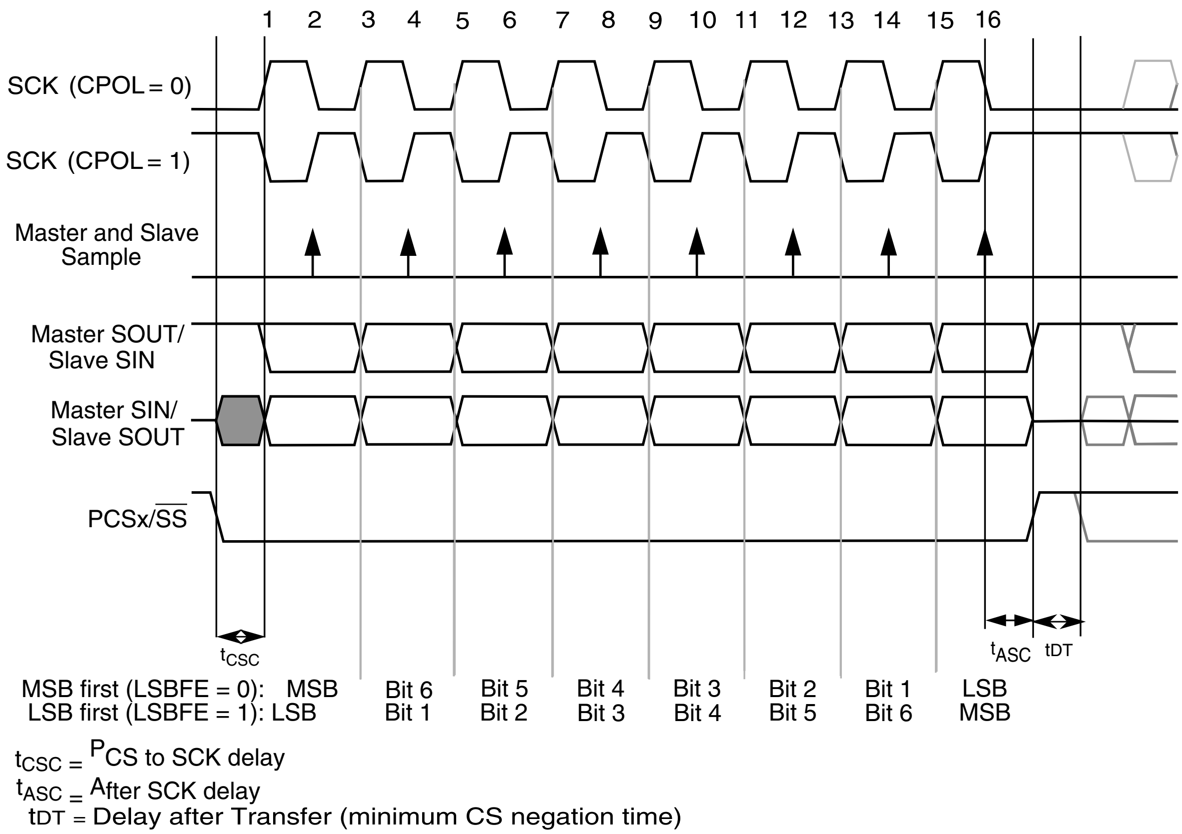


Figure 47-127. Module transfer timing diagram (MTFE=0, CPHA=1, FMSZ=8)

The master initiates the transfer by asserting the PCS signal to the slave. After the t_{CSC} delay has elapsed, the master generates the first SCK edge and at the same time places valid data on the master SOUT pin .The slave responds to the first SCK edge by placing its first data bit on its slave SOUT pin.

At the second edge of the SCK the master and slave sample their SIN pins. For the rest of the frame the master and the slave change the data on their SOUT pins on the odd-numbered clock edges and sample their SIN pins on the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signal. A delay of t_{DT} is inserted before a new frame transfer can be initiated by the master.

47.4.4.3 Continuous Selection Format

Some peripherals must be deselected between every transfer. Other peripherals must remain selected between several sequential serial transfers. The Continuous Selection Format provides the flexibility to handle the following case. The Continuous Selection Format is enabled for the SPI configuration by setting the CONT bit in the SPI command. The behavior of the PCS signals in the configurations is identical so only SPI configuration will be described.

When the CONT bit = 0, the module drives the asserted Chip Select signals to their idle states in between frames. The idle states of the Chip Select signals are selected by the PCSISn bits in the MCR. The following timing diagram is for two four-bit transfers with CPHA = 1 and CONT = 0.

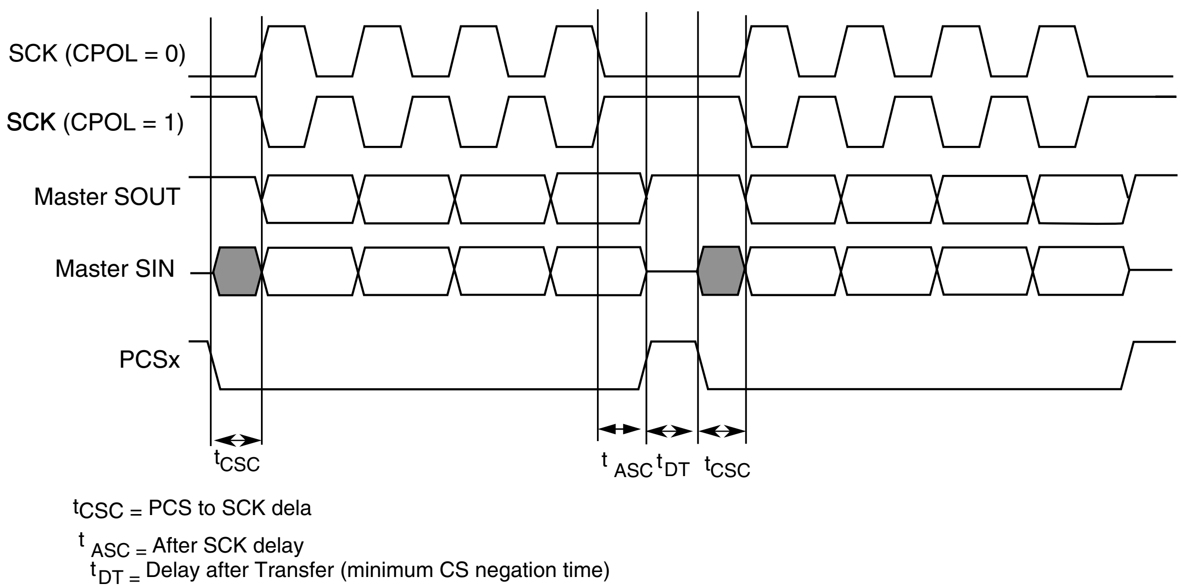


Figure 47-128. Example of non-continuous format (CPHA=1, CONT=0)

When the CONT bit = 1, the PCS signal remains asserted for the duration of the two transfers. The Delay between Transfers (t_{DT}) is not inserted between the transfers. The following figure shows the timing diagram for two four-bit transfers with CPHA = 1 and CONT = 1.

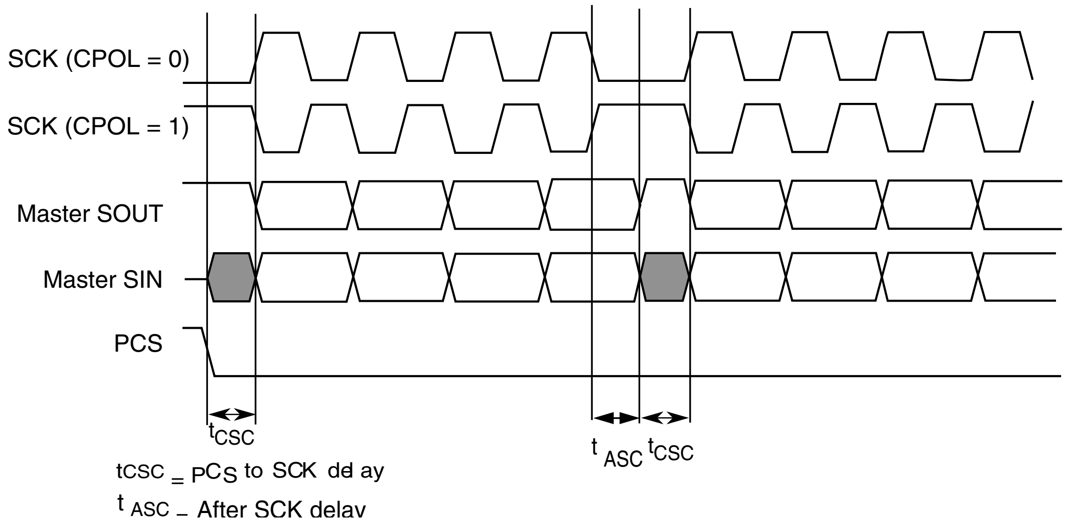


Figure 47-129. Example of continuous transfer (CPHA=1, CONT=1)

When using the module with continuous selection follow these rules:

- All transmit commands must have the same PCSn bits programming.
- The CTARs, selected by transmit commands, must be programmed with the same transfer attributes. Only FMSZ field can be programmed differently in these CTARs.
- When transmitting multiple frames in this mode, the user software must ensure that the last frame has the PUSHR[CONT] bit deasserted in Master mode and the user software must provide sufficient frames in the TX_FIFO to be sent out in Slave mode and the master deasserts the PCSn at end of transmission of the last frame.
- The PUSHR[CONT] bits must be deasserted before asserting MCR[HALT] bit in master mode. This will make sure that the PCSn signals are deasserted. Asserting MCR[HALT] bit during continuous transfer will cause the PCSn signals to remain asserted and hence Slave Device cannot transition from Running to Stopped state.

NOTE

User must fill the TX FIFO with the number of entries that will be concatenated together under one PCS assertion for both master and slave before the TX FIFO becomes empty.

When operating in Slave mode, ensure that when the last entry in the TX FIFO is completely transmitted, that is, the corresponding TCF flag is asserted and TXFIFO is empty, the slave is deselected for any further serial communication; otherwise, an underflow error occurs.

47.4.4.4 Fast Continuous Selection Format

The Fast Continuous Selection Format functions similar to Continuous Selection Format except that the inter command delays, t_{ASC} and t_{CSC}, can be masked out and are not inserted by the hardware.

NOTE

The Fast Continuous Selection Format is available in the SPI configuration only and when Continuous Serial Communication Clock mode is disabled. Masking of delays is not allowed in DSI and CSI configurations or if the transfer is non-continuous.

The Fast Continuous Selection Format is enabled by writing ‘1’ into FCPCS bit of the MCR register. When this bit is asserted, MASC and MCSC bits of the PUSHR register perform the function of mask bits for the transmit frame. These bits individually mask the t_{ASC} and t_{CSC} delays as programmed by the user software. A normal Continuous Selection Format has these two delays for each frame that is transmitted with the CONT bit asserted. In order to avoid these delays and to speed up the transfer process, the software can simply mask these delays while programming the command in the PUSHR register.

While masking the delays, the software must follow the following masking rules, else correct operation is not guaranteed.

- MASC bit masks the “After SCK” delay for the current frame.
- MCSC bit masks the “PCS to SCK” delay for the next frame.
- “After SCK” (t_{ASC}) delay must not be masked when the current frame is the last frame in the continuous selection format.
- The “PCS to SCK” delay for the first frame in the continuous selection format cannot be masked.
- Masking of only t_{ASC} is not allowed. If t_{ASC} is masked then t_{CSC} must be masked too.
- Masking of both t_{ASC} and t_{CSC} delays is allowed. In this case, the delay between two frames is equal to half the baud rate set by the user software.
- Masking of only t_{CSC} is allowed. In this case, the delay between two frames is equal to the t_{ASC} time and thus the user software must ensure that the t_{ASC} time is greater than the baud rate.

- The user software must not mask these delays if the continuous selection format is not used and MCR[FCPCS] is asserted.
- Rules applicable to the Continuous Selection Format are applicable here too.

The following figure shows the timing for a Fast Continuous Selection Format transfer. Here seven frames are transferred with both t_{ASC} and t_{CSC} delays masked except for the last frame that terminated the transfer. The last frame has t_{ASC} delay at its end.

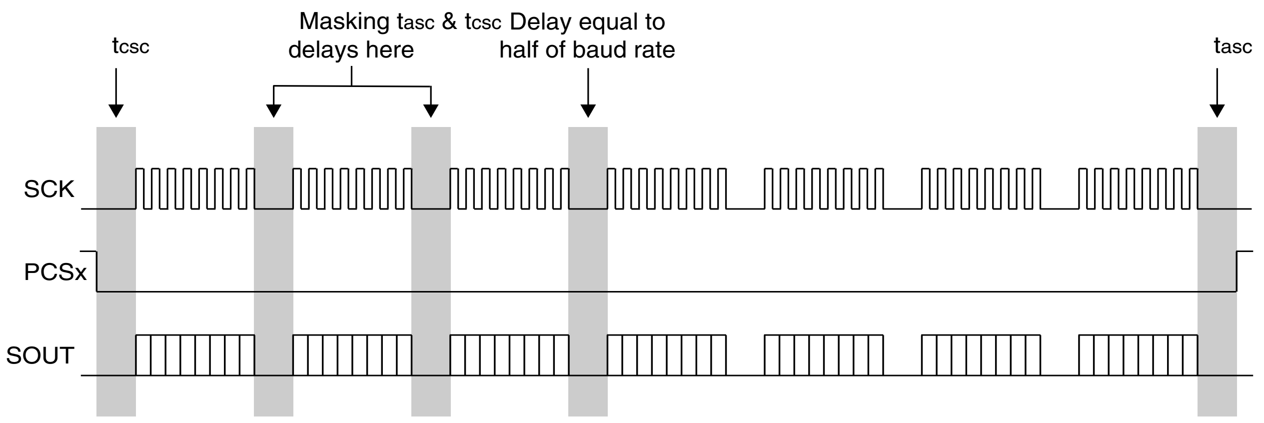


Figure 47-130. Example of Fast Continuous Selection Format

In case any chip select is to be changed, then the fast continuous selection format should be terminated and then the chips selects should change and appropriate delays must be introduced.

47.4.5 Continuous Serial Communications Clock

The module provides the option of generating a Continuous SCK signal for slave peripherals that require a continuous clock.

Continuous SCK is enabled by setting the CONT_SCKE bit in the MCR. Enabling this bit generates the Continuous SCK regardless of the MCR[HALT] bit status. Continuous SCK is valid in all configurations.

Continuous SCK is only supported for CPHA=1. Clearing CPHA is ignored if the CONT_SCKE bit is set. Continuous SCK is supported for Modified Transfer Format.

Clock and transfer attributes for the Continuous SCK mode are set according to the following rules:

- When the module is in SPI configuration, CTAR0 is used initially. At the start of each SPI frame transfer, the CTAR specified by the CTAS for the frame is used.
- In all configurations, the currently selected CTAR remains in use until the start of a frame with a different CTAR specified, or the Continuous SCK mode is terminated.

It is recommended to keep the baud rate the same while using the Continuous SCK. Switching clock polarity between frames while using Continuous SCK can cause errors in the transfer. Continuous SCK operation is not guaranteed if the module is put into the External Stop mode or Module Disable mode.

Enabling Continuous SCK disables the PCS to SCK delay and the Delay after Transfer (t_{DT}) is fixed to one SCK cycle. The following figure is the timing diagram for Continuous SCK format with Continuous Selection disabled.

NOTE

In Continuous SCK mode, for the SPI transfer CTAR0 should always be used, and the TX FIFO must be cleared using the MCR[CLR_TXF] field before initiating transfer.

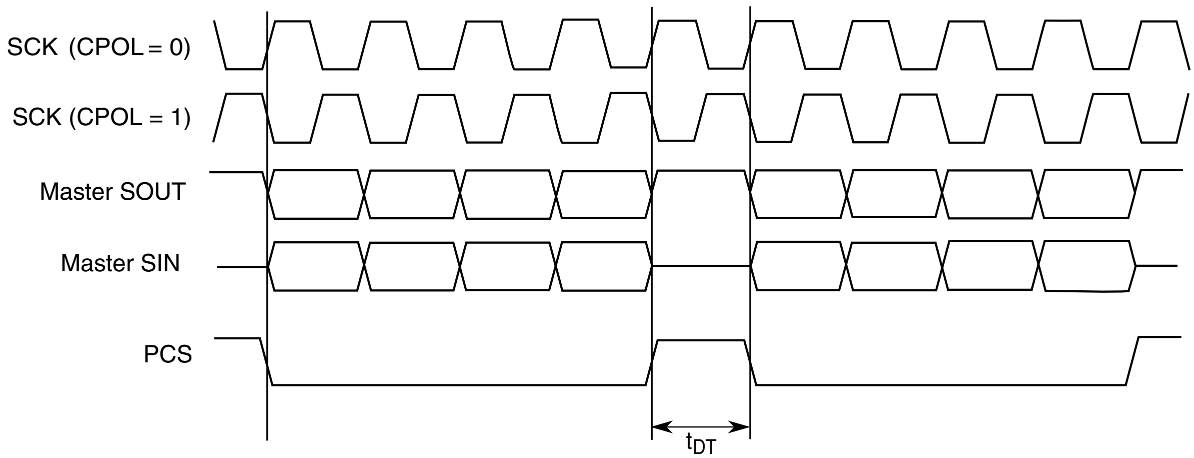


Figure 47-131. Continuous SCK Timing Diagram (CONT=0)

If the CONT bit in the TX FIFO entry is set, PCS remains asserted between the transfers. Under certain conditions, SCK can continue with PCS asserted, but with no data being shifted out of SOUT, that is, SOUT pulled high. This can cause the slave to receive incorrect data. Those conditions include:

- Continuous SCK with CONT bit set, but no data in the TX FIFO.

- Continuous SCK with CONT bit set and entering Stopped state (refer to Start and Stop of module transfers).
- Continuous SCK with CONT bit set and entering Stop mode or Module Disable mode.

The following figure shows timing diagram for Continuous SCK format with Continuous Selection enabled.

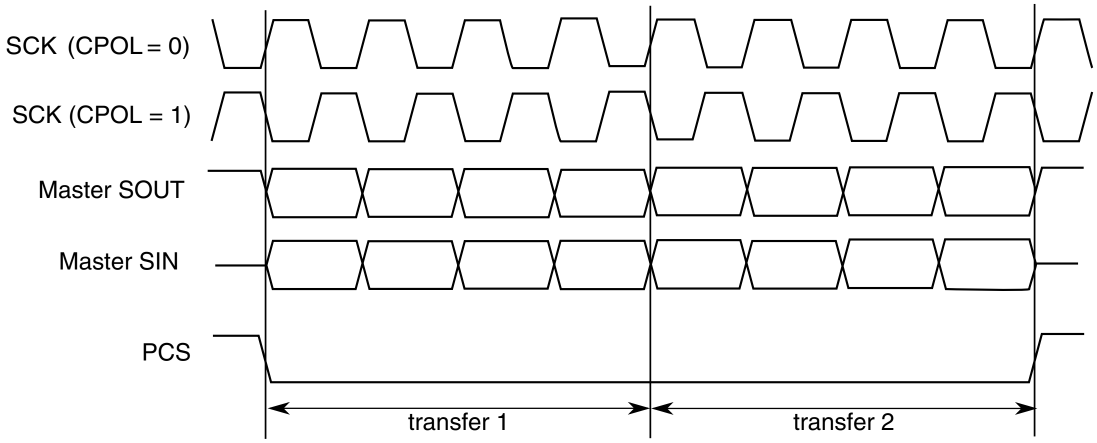


Figure 47-132. Continuous SCK timing diagram (CONT=1)

47.4.6 Slave Mode Operation Constraints

Slave mode logic shift register is buffered. This allows data streaming operation, when the SPI is permanently selected and data is shifted in with a constant rate.

The transmit data is transferred at second SCK clock edge of the each frame to the shift register if the $\overline{SS}$ signal is asserted and any time when transmit data is ready and $\overline{SS}$ signal is negated.

Received data is transferred to the receive buffer at last SCK edge of each frame, defined by frame size programmed to the CTAR0/1 register. Then the data from the buffer is transferred to the RXFIFO or DDR register.

If the $\overline{SS}$ negates before that last SCK edge, the data from shift register is lost.

47.4.7 Parity Generation and Check

The module can generate and check parity in the serial frame. The parity bit replaces the last transmitted bit in the frame. The parity is calculated for all transmitted data bits in frame, not including the last data bit that would be transmitted. The parity generation/

control is done on frame basis. The registers field setting frame size defines the total number of bits in the frame, including the parity bit. Thus, to transmit/receive the same number of data bits with parity check, increase the frame size by one versus the same data size frame without the parity check.

Parity can be selected as odd or even. Parity Errors in the received frame set Parity Error flags in the Status register. The Parity Error Interrupt Requests are generated if enabled. The module can be programmed to stop SPI frame transmission in case of a frame reception with parity error.

47.4.7.1 Parity for SPI Frames

When the module is in the master mode the parity generation is controlled by PE and PP bits of the TX FIFO entries (PUSHR). Setting the PE bit enables parity generation for transmitted SPI frames and parity check for received frames. PP bit defines polarity of the parity bit.

When continuous PCS selection is used to transmit SPI data, two parity generation scenarios are available:

- Generate/check parity for the whole frame
- Generate/check parity for each sub-frame separately.

To generate/check parity for the whole frame set PE bit only in the last command/TX FIFO entry, forming this frame (with the PUSHR register).

To generate/check parity for each sub-frame set PE bit in each command/TX FIFO entry, forming this frame.

If the parity error occurs for received SPI frame, the SR[SPEF] bit is set. If MCR[PES] bit is set, the module stops SPI frames transmission. To resume SPI operation clear the SR[SPEF] or the MCR[PES] bits.

In slave mode the parity is controlled by the PE and PP bits of the CTAR0 register similar to the master mode parity generation without continuous PCS selection.

47.4.8 Interrupts/DMA requests

The module has several conditions that can generate only interrupt requests and two conditions that can generate interrupt or DMA requests. The following table lists these conditions.

Table 47-208. Interrupt and DMA request conditions

Condition	Flag	Interrupt	DMA
End of Queue (EOQ)	EOQF	Yes	-
TX FIFO Fill	TFFF	Yes	Yes
Transfer Complete	TCF	Yes	-
TX FIFO Underflow	TFUF	Yes	-
RX FIFO Drain	RFDF	Yes	Yes
RX FIFO Overflow	RFOF	Yes	-
SPI Parity Error	SPEF	Yes	-

Each condition has a flag bit in the module Status Register (SR) and a Request Enable bit in the DMA/Interrupt Request Select and Enable Register (RSER). Certain flags (as shown in above table) generate interrupt requests or DMA requests depending on configuration of RSER register.

The module also provides a global interrupt request line, which is asserted when any of individual interrupt requests lines is asserted.

47.4.8.1 End of Queue Interrupt Request

The End of Queue Request indicates that the end of a transmit queue is reached. The End of Queue Request is generated when the EOQ bit in the executing SPI command is set and the EOQF_RE bit in the RSER is set.

NOTE

This interrupt request is generated when the last bit of the SPI frame with EOQ bit set is transmitted.

47.4.8.2 Transmit FIFO Fill Interrupt or DMA Request

The Transmit FIFO Fill Request indicates that the TX FIFO is not full. The Transmit FIFO Fill Request is generated when the number of entries in the TX FIFO is less than the maximum number of possible entries, and the TFFF_RE bit in the RSER is set. The TFFF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

NOTE

TFFF flag clears automatically when DMA is used to fill TX FIFO.

To clear TFFF when not using DMA, follow these steps for every PUSH performed using CPU to fill TX FIFO:

1. Wait until TFFF = 1.
2. Write data to PUSHR using CPU.
3. Clear TFFF by writing a 1 to its location. If TX FIFO is not full, this flag will not clear.

47.4.8.3 Transfer Complete Interrupt Request

The Transfer Complete Request indicates the end of the transfer of a serial frame. The Transfer Complete Request is generated at the end of each frame transfer when the TCF_RE bit is set in the RSER.

47.4.8.4 Transmit FIFO Underflow Interrupt Request

The Transmit FIFO Underflow Request indicates that an underflow condition in the TX FIFO has occurred. The transmit underflow condition is detected only for the module operating in Slave mode and SPI configuration . The TFUF bit is set when the TX FIFO of a SPI is empty, and a transfer is initiated from an external SPI master. If the TFUF bit is set while the TFUF_RE bit in the RSER is set, an interrupt request is generated.

47.4.8.5 Receive FIFO Drain Interrupt or DMA Request

The Receive FIFO Drain Request indicates that the RX FIFO is not empty. The Receive FIFO Drain Request is generated when the number of entries in the RX FIFO is not zero, and the RFDF_RE bit in the RSER is set. The RFDF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

47.4.8.6 Receive FIFO Overflow Interrupt Request

The Receive FIFO Overflow Request indicates that an overflow condition in the RX FIFO has occurred. A Receive FIFO Overflow request is generated when RX FIFO and shift register are full and a transfer is initiated. The RFOF_RE bit in the RSER must be set for the interrupt request to be generated.

Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

47.4.8.7 SPI Frame Parity Error Interrupt Request

The SPI Frame Parity Error Flag indicates that a SPI frame with parity error had been received. The SPEF_RE bit in the RSER must be set for the interrupt request to be generated.

47.4.9 Power saving features

The module supports following power-saving strategies:

- External Stop mode
- Module Disable mode – Clock gating of non-memory mapped logic

47.4.9.1 Stop mode (External Stop mode)

The SPI supports the Stop mode protocol. When a request is made to enter External Stop mode, the SPI block acknowledges the request . If a serial transfer is in progress, the SPI waits until it reaches the frame boundary before it is ready to have its clocks shut off . While the clocks are shut off, the SPI memory-mapped logic is not accessible. This also puts the SPI in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state. The states of the interrupt and DMA request signals cannot be changed while in External Stop mode.

47.4.9.2 Module Disable mode

Module Disable mode is a block-specific mode that the module can enter to save power. Host CPU can initiate the Module Disable mode by setting the MDIS bit in the MCR. The Module Disable mode can also be initiated by hardware.

When the MDIS bit is set, the module negates Clock Enable signal at the next frame boundary. Once the Clock Enable signal is negated, it is said to have entered Module Disable Mode. This also puts the module in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state.If implemented, the Clock Enable signal can stop the

clock to the non-memory mapped logic. When Clock Enable is negated, the module is in a dormant state, but the memory mapped registers are still accessible. Certain read or write operations have a different effect when the module is in the Module Disable mode. Reading the RX FIFO Pop Register does not change the state of the RX FIFO. Similarly, writing to the PUSHR Register does not change the state of the TX FIFO. Clearing either of the FIFOs has no effect in the Module Disable mode. Changes to the DIS_TXF and DIS_RXF fields of the MCR have no effect in the Module Disable mode. In the Module Disable mode, all status bits and register flags in the module return the correct values when read, but writing to them has no effect. Writing to the TCR during Module Disable mode has no effect. Interrupt and DMA request signals cannot be cleared while in the Module Disable mode.

47.5 Initialization/application information

This section describes how to initialize the module.

47.5.1 How to manage queues

The queues are not part of the module, but it includes features in support of queue management. Queues are primarily supported in SPI configuration.

1. When module executes last command word from a queue, the EOQ bit in the command word is set to indicate it that this is the last entry in the queue.
2. At the end of the transfer, corresponding to the command word with EOQ set is sampled, the EOQ flag (EOQF) in the SR is set.
3. The setting of the EOQF flag disables serial transmission and reception of data, putting the module in the Stopped state. The TXRXS bit is cleared to indicate the Stopped state.
4. The DMA can continue to fill TX FIFO until it is full or step 5 occurs.
5. Disable DMA transfers by disabling the DMA enable request for the DMA channel assigned to TX FIFO and RX FIFO. This is done by clearing the corresponding DMA enable request bits in the DMA Controller.
6. Ensure all received data in RX FIFO has been transferred to memory receive queue by reading the RXCNT in SR or by checking RFDF in the SR after each read operation of the POPR.
7. Modify DMA descriptor of TX and RX channels for new queues

8. Flush TX FIFO by writing a 1 to the CLR_TXF bit in the MCR. Flush RX FIFO by writing a '1' to the CLR_RXF bit in the MCR.
9. Clear transfer count either by setting CTCNT bit in the command word of the first entry in the new queue or via CPU writing directly to SPI_TCNT field in the TCR.
10. Enable DMA channel by enabling the DMA enable request for the DMA channel assigned to the module TX FIFO, and RX FIFO by setting the corresponding DMA set enable request bit.
11. Enable serial transmission and serial reception of data by clearing the EOQF bit.

47.5.2 Switching Master and Slave mode

When changing modes in the module, follow the steps below to guarantee proper operation.

1. Halt it by setting MCR[HALT].
2. Clear the transmit and receive FIFOs by writing a 1 to the CLR_TXF and CLR_RXF bits in MCR.
3. Set the appropriate mode in MCR[MSTR] and enable it by clearing MCR[HALT].

47.5.3 Initializing Module in Master/Slave Modes

Once the appropriate mode in MCR[MSTR] is configured, the module is enabled by clearing MCR[HALT]. It should be ensured that module Slave is enabled before enabling it's Master. This ensures the Slave is ready to be communicated with, before Master initializes communication.

47.5.4 Baud rate settings

The following table shows the baud rate that is generated based on the combination of the baud rate prescaler PBR and the baud rate scaler BR in the CTARs. The values calculated assume a 100 MHz system frequency and the double baud rate DBR bit is cleared.

Table 47-209. Baud rate values (bps)

		Baud rate divider prescaler values			
		2	3	5	7
Baud Rate Scaler Values	2	25.0M	16.7M	10.0M	7.14M
	4	12.5M	8.33M	5.00M	3.57M
	6	8.33M	5.56M	3.33M	2.38M
	8	6.25M	4.17M	2.50M	1.79M
	16	3.12M	2.08M	1.25M	893k
	32	1.56M	1.04M	625k	446k
	64	781k	521k	312k	223k
	128	391k	260k	156k	112k
	256	195k	130k	78.1k	55.8k
	512	97.7k	65.1k	39.1k	27.9k
	1024	48.8k	32.6k	19.5k	14.0k
	2048	24.4k	16.3k	9.77k	6.98k
	4096	12.2k	8.14k	4.88k	3.49k
	8192	6.10k	4.07k	2.44k	1.74k
	16384	3.05k	2.04k	1.22k	872
	32768	1.53k	1.02k	610	436

47.5.5 Delay settings

The following table shows the values for the Delay after Transfer (t_{DT}) and CS to SCK Delay (T_{CSC}) that can be generated based on the prescaler values and the scaler values set in the CTARs. The values calculated assume a 100 MHz system frequency.

NOTE

The clock frequency mentioned above is given as an example in this chapter. See the clocking chapter for the frequency used to drive this module in the device.

Table 47-210. Delay values

		Delay prescaler values			
		1	3	5	7
Delay scaler values	2	20.0 ns	60.0 ns	100.0 ns	140.0 ns
	4	40.0 ns	120.0 ns	200.0 ns	280.0 ns
	8	80.0 ns	240.0 ns	400.0 ns	560.0 ns
	16	160.0 ns	480.0 ns	800.0 ns	1.1 μs
	32	320.0 ns	960.0 ns	1.6 μs	2.2 μs
	64	640.0 ns	1.9 μs	3.2 μs	4.5 μs
	128	1.3 μs	3.8 μs	6.4 μs	9.0 μs
	256	2.6 μs	7.7 μs	12.8 μs	17.9 μs
	512	5.1 μs	15.4 μs	25.6 μs	35.8 μs
	1024	10.2 μs	30.7 μs	51.2 μs	71.7 μs
	2048	20.5 μs	61.4 μs	102.4 μs	143.4 μs
	4096	41.0 μs	122.9 μs	204.8 μs	286.7 μs
	8192	81.9 μs	245.8 μs	409.6 μs	573.4 μs
	16384	163.8 μs	491.5 μs	819.2 μs	1.1 ms
	32768	327.7 μs	983.0 μs	1.6 ms	2.3 ms
	65536	655.4 μs	2.0 ms	3.3 ms	4.6 ms

47.5.6 Calculation of FIFO pointer addresses

Complete visibility of the TX and RX FIFO contents is available through the FIFO registers, and valid entries can be identified through a memory-mapped pointer and counter for each FIFO. The pointer to the first-in entry in each FIFO is memory mapped. For the TX FIFO the first-in pointer is the Transmit Next Pointer (TXNXTPTR). For the RX FIFO the first-in pointer is the Pop Next Pointer (POPNXTPTR). The following figure illustrates the concept of first-in and last-in FIFO entries along with the FIFO Counter. The TX FIFO is chosen for the illustration, but the concepts carry over to the RX FIFO. See Transmit First In First Out (TX FIFO) buffering mechanism and Receive First In First Out (RX FIFO) buffering mechanism for details on the FIFO operation.

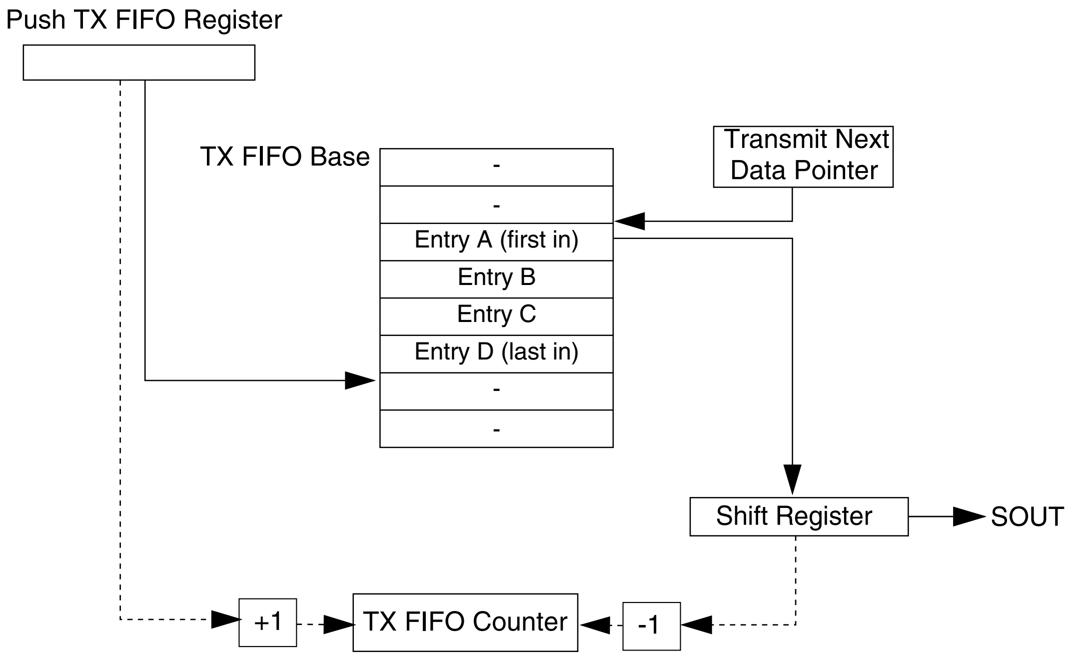


Figure 47-133. TX FIFO pointers and counter

47.5.6.1 Address Calculation for the First-in Entry and Last-in Entry in the TX FIFO

The memory address of the first-in entry in the TX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{TXFIFOBase} + (4 \times \text{TXNXTPTR})$$

The memory address of the last-in entry in the TX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{TXFIFOBase} + 4 \times (\text{TXCTR} + \text{TXNXTPTR} - 1)\text{mod}(\text{TXFIFOdepth})$$

TX FIFO Base - Base address of TX FIFO
TXCTR - TX FIFO Counter
TXNXTPTR - Transmit Next Pointer
TX FIFO Depth - Transmit FIFO depth, implementation specific

47.5.6.2 Address Calculation for the First-in Entry and Last-in Entry in the RX FIFO

The memory address of the first-in entry in the RX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{RX FIFOBase} + (4 \times \text{POPNXTPTR})$$

The memory address of the last-in entry in the RX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{RX FIFO Base} + 4 \times (\text{RXCTR} + \text{POPNXTPTR} - 1) \text{mod} (\text{RXFIFOdepth})$$

RX FIFO Base - Base address of RX FIFO
RXCTR - RX FIFO counter
POPNXTPTR - Pop Next Pointer
RX FIFO Depth - Receive FIFO depth, implementation specific

Chapter 48
Inter-Integrated Circuit (I2C)

48.1 Overview

This chapter describes the Inter-Integrated Circuit (I^2C) bus module implemented on this chip and presents the following topics:

- Introduction to I^2C
- External signal descriptions
- Memory map and register definition
- Functional description
- Initialization/application information

48.2 Introduction to I^2C

This section presents the following topics:

- Definition: I^2C module
- Advantages of the I^2C bus
- Module block diagram
- Features
- Modes of operation
- Definition: I^2C conditions

48.2.1 Definition: I²C module

The I²C module is a functional unit that provides a two-wire— serial data (SDA) and serial clock (SCL) — bidirectional serial bus that provides a simple and efficient method of data exchange between this chip and other devices, such as microcontrollers, EEPROMs, real-time clock devices, analog-to-digital converters, and LCDs.

48.2.2 Advantages of the I²C bus

The synchronous, multiple-master two-wire I²C bus:

- Minimizes interconnections between devices
- Allows the connection of additional devices to the bus for expansion and system development
- Includes collision detection and arbitration that prevent data corruption if two or more masters attempt to control the bus simultaneously
- Does not require an external address decoder

48.2.3 Module block diagram

The following figure shows a block diagram of the I²C module.

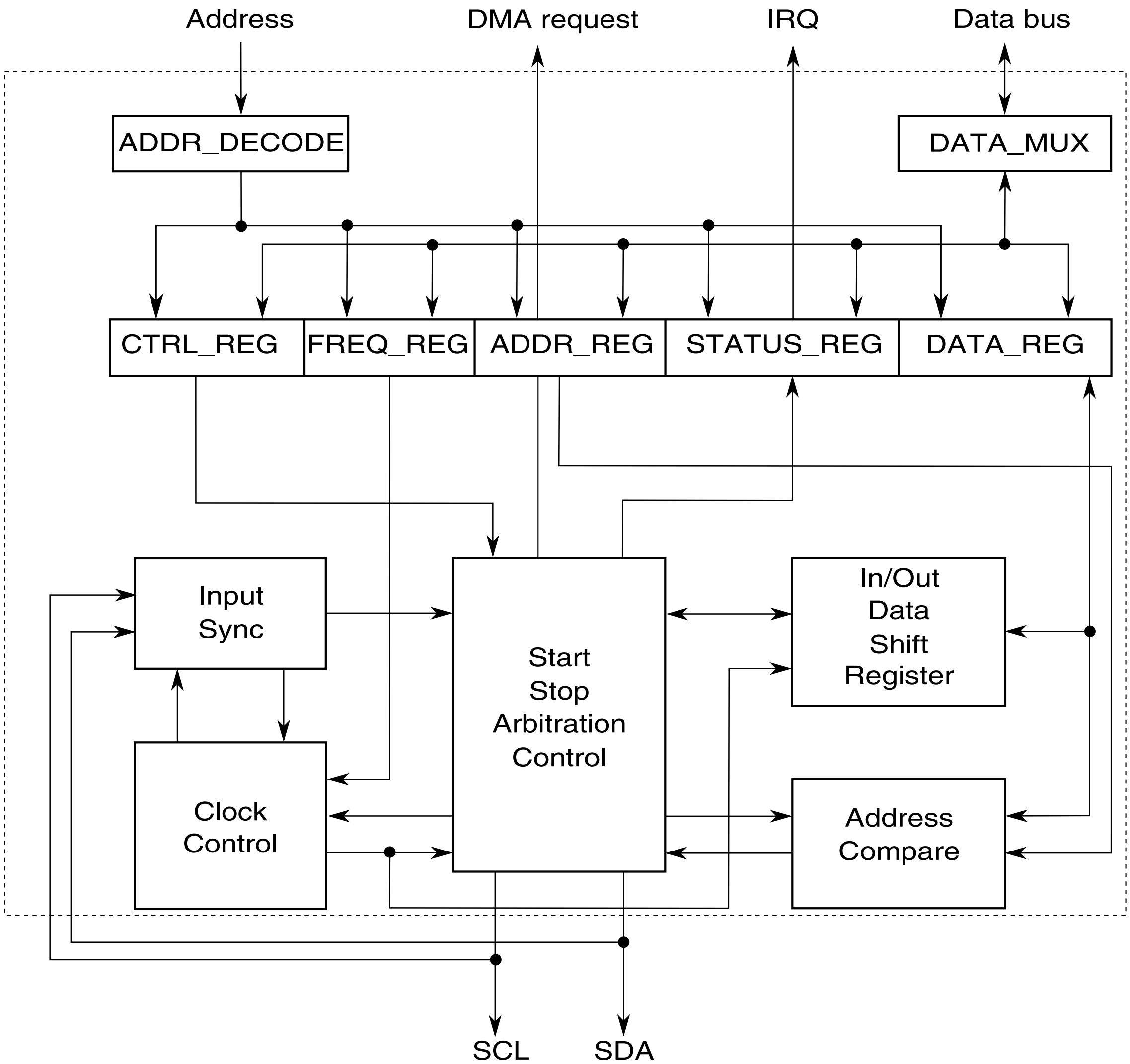


Figure 48-1. I²C block diagram

48.2.4 Features

The I²C module has the following key features:

- Compatible with I²C bus standard[1]
- Operating speeds
  - Up to 100kbps in Standard Mode
  - Up to 400kbps in Fast Mode

1. Compliant with I²C 2.0 standard with the exception that HS (high speed) mode is not supported

    - Operation at higher baud rates (up to a maximum of module clock/20) with reduced bus loading
    - Actual baud rate dependent on the SCL rise time (which depends on external pull-up resistor values and bus loading)
- Multi-master operation
- Software programmable for one of 256 different serial clock frequencies
- Software selectable acknowledge bit
- Interrupt driven byte-by-byte data transfer
- Arbitration lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated start signal generation
- Acknowledge bit generation/detection
- Bus busy detection
- Basic DMA interface
- Maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400pF

48.2.5 Modes of operation

The I^2C module supports the chip modes described in the following table.

Table 48-1. Chip modes supported by the I^2C module

Chip mode	Description	Important notes
RUN	Basic mode of operation	—
STOP	The lowest-power mode that allows the chip to turn of all the clocks to the I^2C module	The I^2C module can enter this mode when there are no active transfers (active data between valid START and STOP conditions) on the bus.
IPG DEBUG	Allows the chip to freeze all ongoing activities (such as an ongoing transaction, counter values, and register status) for debugging	See IPG DEBUG mode.

In addition to chip modes, the I2C module has several module-specific modes. These are described in the following table.

Table 48-2. Module-specific modes supported by the I²C module

Module mode	Description	Important notes
Master mode	The I²C module is the driver of the SDA line.	• Do not use the I²C module's slave address as a calling address. • The I²C module cannot be a master and a slave simultaneously.
Slave mode	The I²C module is not the driver of the SDA line.	• Enable the I²C module before a START condition from a non-I²C master is detected. • By default the I²C module performs as a slave receiver.

48.2.6 Definition: I²C conditions

The following table shows the I²C-specific conditions defined for the I²C module.

Table 48-3. I²C Conditions

Condition	Description
START	A condition that denotes the beginning of a new data transfer and awakens all slaves. Each data transfer contains several bytes of data. It is defined as a high-to-low transition of SDA while SCL is high, as shown in the following figure.
STOP	A condition generated by the master to terminate a transfer and free the bus. It is defined as a low-to-high transition of SDA while SCL is high, as shown in the following figure.
Repeated START	A START condition that is generated without a STOP condition to terminate the previous transfer.

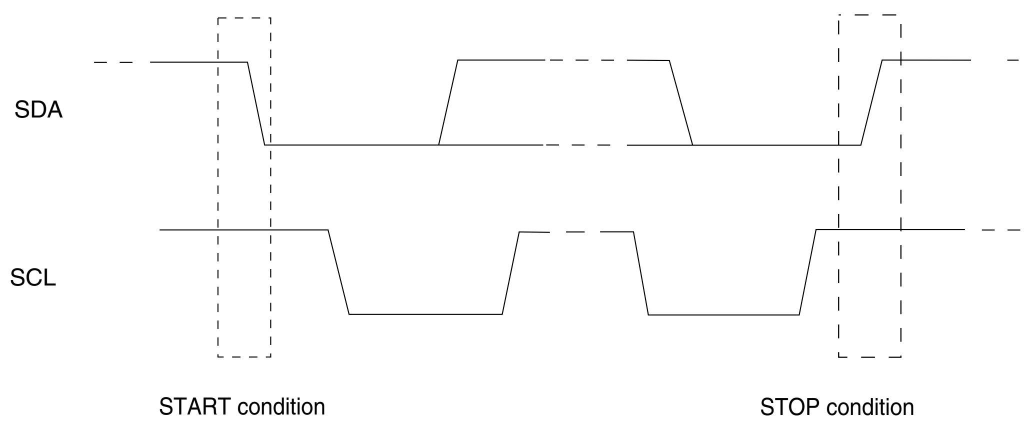


Figure 48-2. START and STOP conditions

48.3 External signal descriptions

This section presents the following topics:

- Signal overview
- Detailed external signal descriptions

48.3.1 Signal overview

The I^2C module uses the Serial Data (SDA) and Serial Clock (SCL) signals as a communication interconnect with other devices. The signal patterns driven on the SDA signal represent address, data, or read/write information at different stages of the protocol.

All devices connected to the SDA and SCL signals must have open-drain or open-collector outputs. The logical AND function is performed on both signals with external pull-up resistors. For the electrical characteristics of these signals, see the data sheet for this chip.

48.3.2 Detailed external signal descriptions

The SDA and SCL signals are described in the following table.

Table 48-4. External signal descriptions

Signal	Description
SCL	Bidirectional serial clock line of the module, compatible with the I^2C bus specification
SDA	Bidirectional serial data line of the module, compatible with the I^2C bus specification

48.4 Memory map and register definition

48.4.1 Register accessibility

All registers are accessible via 8-bit, 16-bit, or 32-bit accesses. However, 16-bit accesses must be aligned to 16-bit boundaries, and 32-bit accesses must be aligned to 32-bit boundaries.

As an example, the IBFD is at address offset 0x01, and can be accessed in any of the following ways:

- 8-bit R/W access of address offset 0x0001
- Second byte of 16-bit R/W access of address offset 0x0000
- Second byte of 32-bit R/W access of address offset 0x0000

The IBFD register cannot be accessed by a 16- or 32-bit RW operation at address offset 0x0001 because those operations require an address aligned to a 16- or 32-bit boundary.

48.4.2 Register figure conventions

The register figures show the field structure using the conventions in the following figure.

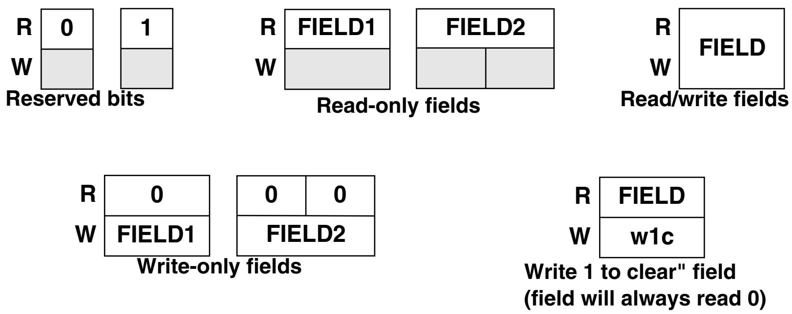


Figure 48-3. Register figure convention

The memory map for the I^{2}C module is given below. The total address for each register is the sum of the base address for the I^{2}C module and the address offset for each register.

I2C memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_6000	I2C Bus Address Register (I2C0_IBAD)	8	R/W	00h	48.4.3/2588
4006_6001	I2C Bus Frequency Divider Register (I2C0_IBFD)	8	R/W	00h	48.4.4/2589
4006_6002	I2C Bus Control Register (I2C0_IBCR)	8	R/W	80h	48.4.5/2589
4006_6003	I2C Bus Status Register (I2C0_IBSR)	8	R/W	80h	48.4.6/2591
4006_6004	I2C Bus Data I/O Register (I2C0_IBDR)	8	R/W	00h	48.4.7/2592
4006_6005	I2C Bus Interrupt Config Register (I2C0_IBIC)	8	R/W	00h	48.4.8/2593
4006_6006	I2C Bus Debug Register (I2C0_IBDBG)	8	R/W	00h	48.4.9/2594
4006_7000	I2C Bus Address Register (I2C1_IBAD)	8	R/W	00h	48.4.3/2588

Table continues on the next page...

I2C memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_7001	I2C Bus Frequency Divider Register (I2C1_IBFD)	8	R/W	00h	48.4.4/2589
4006_7002	I2C Bus Control Register (I2C1_IBCR)	8	R/W	80h	48.4.5/2589
4006_7003	I2C Bus Status Register (I2C1_IBSR)	8	R/W	80h	48.4.6/2591
4006_7004	I2C Bus Data I/O Register (I2C1_IBDR)	8	R/W	00h	48.4.7/2592
4006_7005	I2C Bus Interrupt Config Register (I2C1_IBIC)	8	R/W	00h	48.4.8/2593
4006_7006	I2C Bus Debug Register (I2C1_IBDBG)	8	R/W	00h	48.4.9/2594
400E_6000	I2C Bus Address Register (I2C2_IBAD)	8	R/W	00h	48.4.3/2588
400E_6001	I2C Bus Frequency Divider Register (I2C2_IBFD)	8	R/W	00h	48.4.4/2589
400E_6002	I2C Bus Control Register (I2C2_IBCR)	8	R/W	80h	48.4.5/2589
400E_6003	I2C Bus Status Register (I2C2_IBSR)	8	R/W	80h	48.4.6/2591
400E_6004	I2C Bus Data I/O Register (I2C2_IBDR)	8	R/W	00h	48.4.7/2592
400E_6005	I2C Bus Interrupt Config Register (I2C2_IBIC)	8	R/W	00h	48.4.8/2593
400E_6006	I2C Bus Debug Register (I2C2_IBDBG)	8	R/W	00h	48.4.9/2594
400E_7000	I2C Bus Address Register (I2C3_IBAD)	8	R/W	00h	48.4.3/2588
400E_7001	I2C Bus Frequency Divider Register (I2C3_IBFD)	8	R/W	00h	48.4.4/2589
400E_7002	I2C Bus Control Register (I2C3_IBCR)	8	R/W	80h	48.4.5/2589
400E_7003	I2C Bus Status Register (I2C3_IBSR)	8	R/W	80h	48.4.6/2591
400E_7004	I2C Bus Data I/O Register (I2C3_IBDR)	8	R/W	00h	48.4.7/2592
400E_7005	I2C Bus Interrupt Config Register (I2C3_IBIC)	8	R/W	00h	48.4.8/2593
400E_7006	I2C Bus Debug Register (I2C3_IBDBG)	8	R/W	00h	48.4.9/2594

48.4.3 I2C Bus Address Register (I2Cx_IBAD)

This register contains the address the I^{2}C Bus will respond to when addressed as a slave. This is not the address sent on the bus during the address transfer.

Address: Base address + 0h offset

Bit	7	6	5	4	3	2	1	0
Read	ADR							0
Write								
Reset	0	0	0	0	0	0	0	0

I2Cx_IBAD field descriptions

Field	Description
7–1 ADR	Slave Address. Specific slave address to be used by the I^{2}C Bus module. **NOTE:** The default mode of I^{2}C Bus is slave mode for an address match on the bus.

Table continues on the next page...

I2Cx_IBAD field descriptions (continued)

Field	Description
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

48.4.4 I2C Bus Frequency Divider Register (I2Cx_IBFD)

Address: Base address + 1h offset

Bit	7	6	5	4	3	2	1	0
Read Write	IBC							
Reset	0	0	0	0	0	0	0	0

I2Cx_IBFD field descriptions

Field	Description
7–0 IBC	I-Bus Clock Rate. This field is used to prescale the clock for bit rate selection. The bit clock generator is implemented as a prescale divider. The IBC bits are decoded to give the Tap and Prescale values as follows: 0-1 Selects the prescaled shift register (see Clock rate and IBFD settings) 2-4 Selects the prescaler divider (see Clock rate and IBFD settings) 5-7 Selects the shift register tap point (see Clock rate and IBFD settings)

48.4.5 I2C Bus Control Register (I2Cx_IBCR)

Address: Base address + 2h offset

Bit	7	6	5	4	3	2	1	0
Read	MDIS	IBIE	MSSL	TXRX	NOACK	0	DMAEN	0
Write						RSTA		
Reset	1	0	0	0	0	0	0	0

I2Cx_IBCR field descriptions

Field	Description
7 MDIS	Module disable. This bit controls the software reset of the entire I2C Bus module. **NOTE:** If the I2C Bus module is enabled in the middle of a byte transfer, the interface behaves as follows: slave mode ignores the current transfer on the bus and starts operating whenever a subsequent start condition is detected. Master mode will not be aware that the bus is busy, hence if a start cycle is initiated then the current bus cycle may become corrupt. This would ultimately result in either the current bus master or the I2C Bus module losing arbitration, after which, bus operation would return to normal.

Table continues on the next page...

I2Cx_IBCR field descriptions (continued)

Field	Description
	0 The I^2C Bus module is enabled. This bit must be cleared before any other IBCR bits have any effect 1 The module is reset and disabled. This is the power-on reset situation. When high, the interface is held in reset, but registers can still be accessed. Status register bits (IBSR) are not valid when module is disabled.
6 IBIE	I-Bus Interrupt Enable. 0 Interrupts from the I^2C Bus module are disabled. This does not clear any currently pending interrupt condition. 1 Interrupts from the I^2C Bus module are enabled. An I^2C Bus interrupt occurs provided the IBIF bit in the status register is also set.
5 MSSL	Master/Slave mode select. When this bit is changed from 0 to 1, a START signal is generated on the bus and the master mode is selected. When this bit is changed from 1 to 0, a STOP signal is generated and the operation mode changes from master to slave. A STOP signal should be generated only if the IBIF flag is set. This field is cleared without generating a STOP signal when the master loses arbitration. 0 Slave Mode 1 Master Mode
4 TXRX	Transmit/Receive mode select. This bit selects the direction of master and slave transfers. When addressed as a slave this bit should be set by software according to the SRW bit in the status register. In master mode this bit should be set according to the type of transfer required. Therefore, for address cycles, this bit will always be high. 0 Receive 1 Transmit
3 NOACK	Data Acknowledge disable. This bit specifies the value driven onto SDA during data acknowledge cycles for both master and slave receivers. The I^2C module will always acknowledge address matches, provided it is enabled, regardless of the value of NOACK. **NOTE:** Values written to this bit are only used when the I^2C Bus is a receiver, not a transmitter. 0 An acknowledge signal will be sent out to the bus at the 9th clock bit after receiving one byte of data 1 No acknowledge signal response is sent (i.e., acknowledge bit = 1)
2 RSTA	Repeat Start. Writing a one to this bit will generate a repeated START condition on the bus, provided it is the current bus master. This bit will always be read as a low. Attempting a repeated start at the wrong time, if the bus is owned by another master, will result in loss of arbitration. 0 No effect 1 Generate repeat start cycle
1 DMAEN	DMA Enable. When this bit is set, the DMA Tx and Rx lines will be asserted when the I^2C module requires data to be read or written to the data register. No Transfer Done interrupts will be generated when this bit is set, however an interrupt will be generated if the loss of arbitration or addressed as slave conditions occur. The DMA mode is only valid when the I^2C module is configured as a Master and the DMA transfer still requires CPU intervention at the start and the end of each frame of data. See the DMA Application Information section for more details. 0 Disable the DMA TX/RX request signals 1 Enable the DMA TX/RX request signals
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

48.4.6 I2C Bus Status Register (I2Cx_IBSR)

Address: Base address + 3h offset

Bit	7	6	5	4	3	2	1	0
Read	TCF	IAAS	IBB	IBAL	0	SRW	IBIF	RXAK
Write				w1c			w1c	
Reset	1	0	0	0	0	0	0	0

I2Cx_IBSR field descriptions

Field	Description
7 TCF	Transfer complete. While one byte of data is being transferred, this bit is cleared. It is set by the falling edge of the 9th clock of a byte transfer. **NOTE:** This bit is only valid during or immediately following a transfer to the I^2C module or from the I^2C module. 0 Transfer in progress 1 Transfer complete
6 IAAS	Addressed as a slave. When its own specific address (I-Bus Address Register) is matched with the calling address, this bit is set. The CPU is interrupted provided the IBIE is set. Then the CPU needs to check the SRW bit and set the TXRX field accordingly. Writing to the I-Bus Control Register clears this bit. 0 Not addressed 1 Addressed as a slave
5 IBB	Bus busy. This bit indicates the status of the bus. When a START signal is detected, IBB is set. If a STOP signal is detected, IBB is cleared and the bus enters idle state. 0 Bus is Idle 1 Bus is busy
4 IBAL	Arbitration Lost. The arbitration lost bit (IBAL) is set by hardware when the arbitration procedure is lost. Arbitration is lost in the following circumstances: • SDA is sampled low when the master drives a high during an address or data transmit cycle. • SDA is sampled low when the master drives a high during the acknowledge bit of a data receive cycle. • A start cycle is attempted when the bus is busy. • A repeated start cycle is requested in slave mode. • A stop condition is detected when the master did not request it. This bit must be cleared by software, by writing a one to it. A write of zero has no effect.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 SRW	Slave Read/Write. When the IAAS bit is set, this bit indicates the value of the R/W command bit of the calling address sent from the master. This bit is only valid when the I-Bus is in slave mode, a complete address transfer has occurred with an address match and no other transfers have been initiated. By

Table continues on the next page...

I2Cx_IBSR field descriptions (continued)

Field	Description
	reading this field, the CPU can detect slave transmit/receive mode according to the command of the master. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IBIF	I-Bus Interrupt Flag. The IBIF bit is set when one of the following conditions occurs: • Arbitration lost (IBAL bit set) • Byte transfer complete (TCF bit set and DMAEN bit not set) • Addressed as slave (IAAS bit set) • NoAck from Slave (MS & Tx bits set) • I^2C Bus going idle (IBB high-low transition and enabled by BIIE) A processor interrupt request will be caused if the IBIE bit is set. This bit must be cleared by software, by writing a one to it. A write of zero has no effect on this bit. In DMA mode (DMAEN set) a byte transfer complete condition will not trigger the setting of IBIF. All other conditions still apply.
0 RXAK	Received Acknowledge. This is the value of SDA during the acknowledge bit of a bus cycle. If the received acknowledge bit (RXAK) is low, it indicates an acknowledge signal has been received after the completion of 8 bits data transmission on the bus. If RXAK is high, it means no acknowledge signal is detected at the 9th clock. This bit is valid only after transfer is complete. 0 Acknowledge received 1 No acknowledge received

48.4.7 I2C Bus Data I/O Register (I2Cx_IBDR)

In master transmit mode, when data is written to the IBDR, a data transfer is initiated. The most significant bit is sent first. In master receive mode, reading this register initiates next byte data receiving. In slave mode, the same functions are available after an address match has occurred.

NOTE

The IBCR[TXRX] field must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For instance, if the I^2C is configured for master transmit but a master receive is desired, then reading the IBDR will not initiate the receive.

Reading the IBDR will return the most recent byte received while the I^2C is configured in either master receive or slave receive modes. The IBDR does not reflect every byte that is transmitted on the I^2C bus, nor can software verify that a byte has been written to the IBDR correctly by reading it back.

In master transmit mode, the first byte of data written to IBDR following assertion of MSSL is used for the address transfer and should comprise the calling address (in position DATA[7:1]) concatenated with the required R/ $\overline{W}$ bit (in position D0).

Address: Base address + 4h offset

Bit	7	6	5	4	3	2	1	0
Read Write	DATA							
Reset	0	0	0	0	0	0	0	0

I2Cx_IBDR field descriptions

Field	Description
7–0 DATA	Data transmitted or received

48.4.8 I2C Bus Interrupt Config Register (I2Cx_IBIC)

Address: Base address + 5h offset

Bit	7	6	5	4	3	2	1	0
Read	BIIE	0			0	0	0	0
Write								
Reset	0	0	0	0	0	0	0	0

I2Cx_IBIC field descriptions

Field	Description
7 BIIE	Bus Idle Interrupt Enable bit. This config bit can be used to enable the generation of an interrupt once the I^{2}C bus becomes idle. Once this bit is set, an IBB high-low transition will set the IBIF bit. This feature can be used to signal to the CPU the completion of a STOP on the I^{2}C bus. 0 Bus Idle Interrupts disabled 1 Bus Idle Interrupts enabled
6–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

48.4.9 I2C Bus Debug Register (I2Cx_IBDBG)

Address: Base address + 6h offset

Bit	7	6	5	4
Read	0			
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	0		IPG_DEBUG_HALTED	IPG_DEBUG_EN
Write				
Reset	0	0	0	0

I2Cx_IBDBG field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 IPG_DEBUG_ HALTED	Debug Halted Bit: This is s status bit which can be read back after asserting the debug enable signal so as to know if the IP has entered the debug mode or not. 0 IP is still executing a transaction 1 IP has entered the debug mode
0 IPG_DEBUG_EN	Debug enable bit. This bit is used to enable IP enter the debug mode provided the IPG DEBUG signal is high. All the registers, counter values and status bits are frozen and can be accessed by the CPU. **NOTE:** If the assertion of this bit along with the IPG DEBUG signal happens in the middle of a transaction, IP enters the debug mode only after successful completion of the current transaction after which no further transaction can take place until the debug mode is exited. 0 Normal operation, Bus Idle Interrupts disabled 1 IP is in debug mode

48.5 Functional description

This section presents the following topics:

- Notes about module operation
- Transactions
- Arbitration procedure
- Clock behavior
- Interrupts

- IPG DEBUG mode
- DMA interface

48.5.1 Notes about module operation

- The I^2C module always performs as a slave receiver by default, unless explicitly programmed to be a master or slave transmitter.
- When the I^2C module is acting as a master, it must not try to call its own slave address.

48.5.2 Transactions

This section presents the following topics:

- Protocol overview
- Transaction protocol definitions
- High-level protocol steps
- START condition
- Slave address transmission
- Data transmission
- STOP condition
- Repeated START condition

48.5.2.1 Protocol overview

The following figure shows the behavior of SCL and SDA during a typical I^2C transaction.

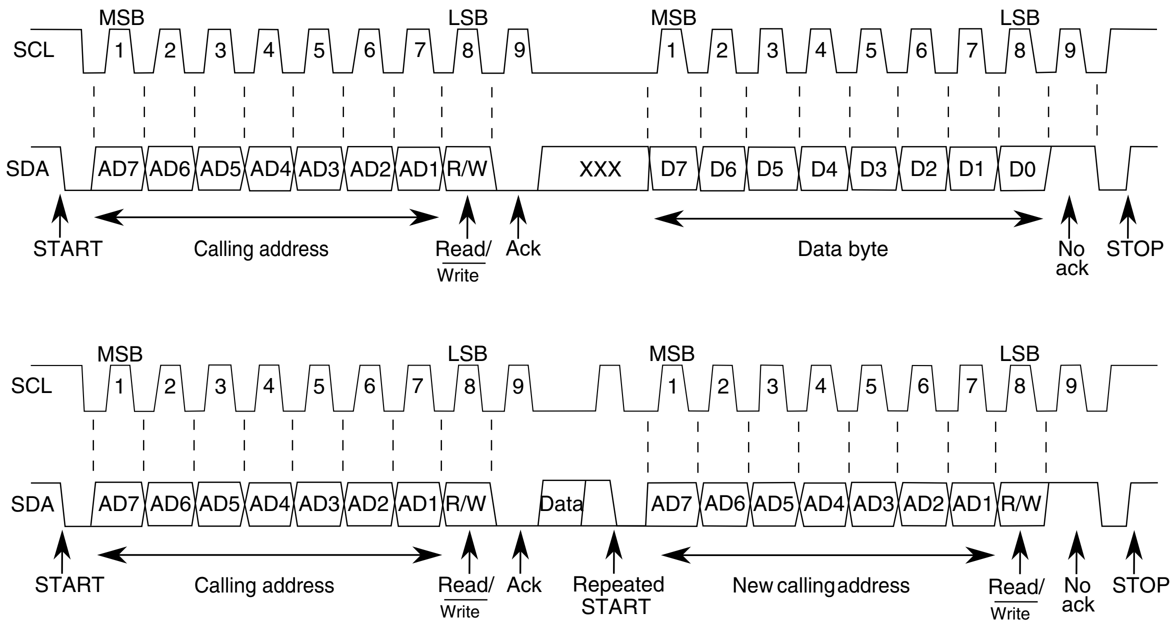


Figure 48-39. I²C transaction protocol

48.5.2.2 Transaction protocol definitions

This section defines several important terms presented in Figure 48-39.

Table 48-45. I²C definitions

Term	Definition
START	A START condition, as defined in Section 1.2.6, Definition: I²C conditions"
STOP	A STOP condition, as defined in Section 1.2.6, Definition: I²C conditions"
Calling (slave) address	A seven-bit address used to identify a slave on the I²C bus. The requirements for specifying this address are presented in Section 1.5.2.3, I²C calling address requirements."
Read/write (R/W)	A bit that specifies the direction of the data transfer to the slave as follows: • 0=The data is being transferred from the master to the slave ("write") • 1=The data is being transferred from the slave to the master ("read")
Ack	A bit that specifies the acknowledgement of a calling address, indicated by pulling SDA low.

48.5.2.3 I²C calling address requirements

The calling addresses of the devices used on an I²C network are subject to the following requirements:

- Each slave must have a unique calling address.
- A master must not transmit a calling address that is the same as its own slave address.

48.5.2.4 High-level protocol steps

The I^2C protocol conceptually supports two types of transfers, which are illustrated in Figure 48-39. The significant steps in these transfers are presented in the following table. Details of each of these steps are presented in subsequent sections.

Table 48-46. I^2C high-level protocol steps

Standard Transfer	Repeated START Transfer
1. START condition	1. START condition
2. Slave target or general call address transmission	2. Slave target or general call address transmission
3. Acknowledgment from slave	3. Acknowledgment from slave
4. Data transfer	4. Data transfer
5. STOP condition	5. Repeated START condition
6. (repeat Steps 1–4)	6. (repeat Steps 2–4 as needed)
	7. STOP condition.
	8. (repeat Steps 1–7)

48.5.2.5 START condition

When the bus is free, that is, no master device is engaging the bus (both SCL and SDA lines are at logical high), a master may initiate communication by sending a START condition (see Definition: I^2C conditions). This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and brings all slaves out of their idle states.

48.5.2.6 Slave address transmission

The master transmits the slave address on the next clock cycle after the START condition (see START condition). The process of slave address transmission is presented in the following table.

Table 48-47. Slave address transmission process

Step	Action
1	The master transmits the seven-bit slave address.
2	The master transmits the R/W bit.
3	Each slave examines the transmitted address and compares it to its own. If the addresses match, the slave device returns the acknowledge bit on the ninth SCL clock cycle.
4	The master waits for the acknowledge bit and determines the next step as follows: • The acknowledge bit is set: The master must initiate a data transfer followed by either a STOP condition or a repeated START condition. • The acknowledge bit is cleared: The master must wait for SCL to return to logic zero.

48.5.2.7 Data transmission

A data transfer session has the following characteristics:

- Can transmit one or more bytes of data
- Awakens all slaves
- Proceeds on a byte-by-byte basis in the direction specified by the R/W bit sent by the calling master

The transmitted data is subject to the following requirements:

- Each data byte must consist of 8 bits.
- Data bits can be changed only while SCL is low and must be held stable while SCL is high.
- One data bit is transmitted during one SCL clock pulse.
- The most significant bit (msb) must be transmitted first.
- Each data byte must be followed by an acknowledge bit on the ninth SCL clock pulse.

If the slave receiver does not acknowledge the master, the SDA line must be left high by the slave. The master can then generate a stop condition to abort the data transfer or a START condition (repeated START) to begin a new calling.

If the master receiver does not acknowledge the slave transmitter after a byte of transmission, the slave interprets that the end-of-data has been reached. Then the slave releases the SDA line for the master to generate a STOP or a START condition.

48.5.2.8 STOP condition

The master can terminate the communication by generating a STOP condition (see Definition: I^{2}C conditions). It can do so even if the slave has generated an acknowledge, at which point the slave must release the bus.

A master is not required to send a STOP condition at the end of every transfer. For more information, see Repeated START condition.

48.5.2.9 Repeated START condition

The I^{2}C protocol also supports a repeated START condition, which can be generated without a preceding STOP condition. A master device can use this condition to communicate with another slave or with the same slave in a different mode without releasing the bus. This condition is illustrated in the second timing diagram of Figure 48-39.

48.5.3 Arbitration procedure

The I^{2}C bus is a true multi-master bus that allows more than one master to be connected to it. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock, for which the low period is equal to the longest clock low period and the high is equal to the shortest one among the masters. The relative priority of the contending masters is determined by a data arbitration procedure. A bus master loses arbitration if it transmits logic "1" while another master transmits logic "0". The losing masters immediately switch over to slave receive mode and stop driving the SDA output. In this case, the transition from master to slave mode does not generate a STOP condition. Meanwhile, a status bit is set by hardware to indicate loss of arbitration.

48.5.4 Clock behavior

This section presents the following topics:

- Clock synchronization
- Clock stretching

- Handshaking
- Clock rate and IBFD settings

48.5.4.1 Clock synchronization

Due to the wired-AND logic on the SCL line, a high-to-low transition on the SCL line affects all devices connected on the bus. The devices begin counting their low period when the master drives the SCL line low. After a device has driven SCL low, it holds the SCL line low until the clock high state is reached.

However, the change of low-to-high in a device clock may not change the state of the SCL line if another device is still within its low period. Therefore, the synchronized clock signal, SCL, is held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time (see the following figure). When all devices concerned have counted off their low period, the synchronized SCL line is released and pulled high. Then there is no difference between the devices' clocks and the state of the SCL line, and all the devices begin counting their high periods. The first device to complete its high period pulls the SCL line low again.

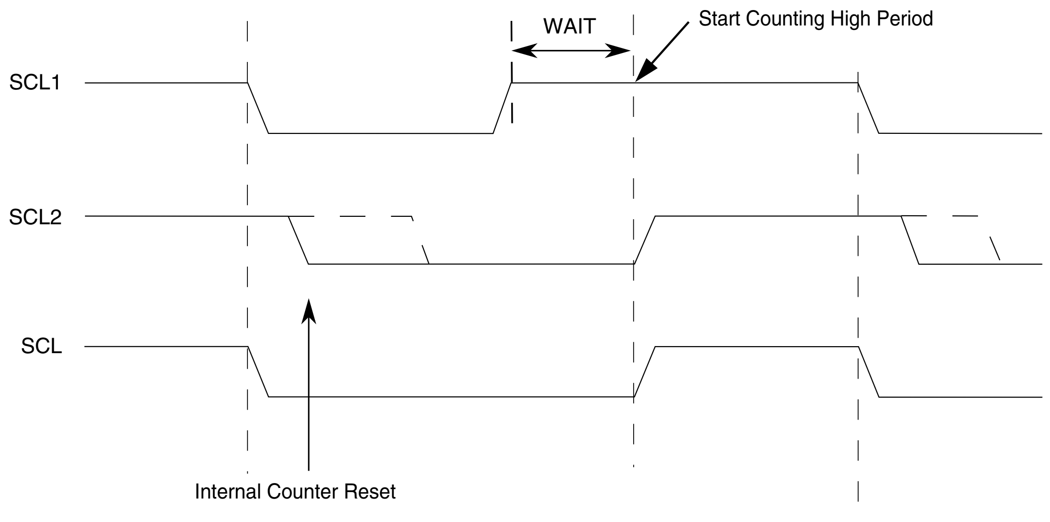


Figure 48-40. I²C bus clock synchronization

48.5.4.2 Clock stretching

The clock synchronization mechanism can be used by slaves to slow down the bit rate of a transfer. After the master has driven SCL low, the slave can drive SCL low for the required period and then release it. If the slave SCL low period is greater than the master SCL low period then the resulting SCL bus signal low period is stretched.

48.5.4.3 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfer. Slave devices may hold the SCL low after completion of one byte transfer (9 bits). In such cases, it halts the bus clock and forces the master clock into wait state until the slave releases the SCL line.

48.5.4.4 Clock rate and IBFD settings

The following tables describe the settings of several fields in the IBFD register (see I2C Bus Frequency Divider Register (I2C_IBFD)).

Table 48-48. I-Bus multiplier factor

IBC[7-6]	MUL
00	01
01	02
10	04
11	Reserved

Table 48-49. I-Bus prescaler divider values

IBC[5-3]	scl2start (clocks)	scl2stop (clocks)	scl2tap (clocks)	tap2tap (clocks)
000	2	7	4	1
001	2	7	4	2
010	2	9	6	4
011	6	9	6	8
100	14	17	14	16
101	30	33	30	32
110	62	65	62	64
111	126	129	126	128

Table 48-50. I-Bus tap and prescale values

IBC[2-0]	SCL Tap (clocks)	SDA Tap (clocks)
000	5	1
001	6	1
010	7	2

Table continues on the next page...

Table 48-50. I-Bus tap and prescale values (continued)

IBC[2-0]	SCL Tap (clocks)	SDA Tap (clocks)
011	8	2
100	9	3
101	10	3
110	12	4
111	15	4

The number of clocks from the falling edge of SCL to the first tap (Tap[1]) is defined by the values shown in the scl2tap column of Table 48-49. All subsequent tap points are separated by 2^{IBC} as shown in the tap2tap column in Table 48-49. The SCL Tap is used to generate the SCL period and the SDA Tap is used to determine the delay from the falling edge of SCL to the change of state of SDA that is the SDA Hold time.

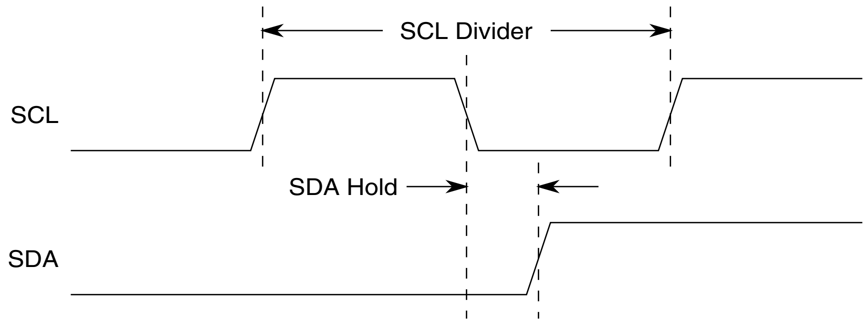


Figure 48-41. SDA Hold time

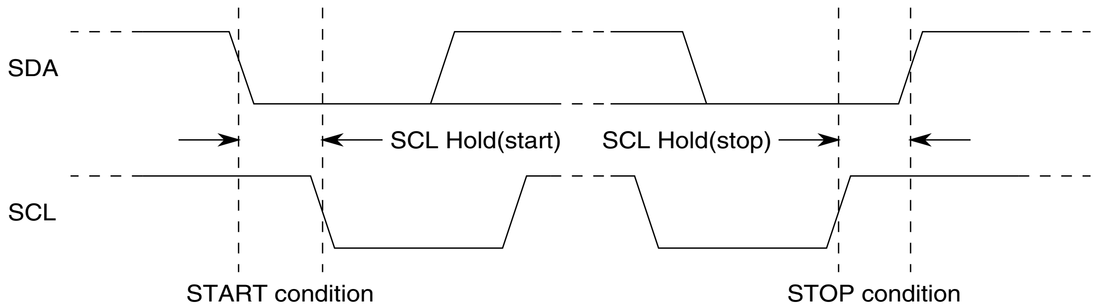


Figure 48-42. SCL Divider and SDA Hold

The equation used to generate the divider values from the IBFD bits is:

$$\text{SCL Divider} = \text{MUL} \times \{2 \times (\text{scl2tap} + [(\text{SCL\_Tap} - 1) \times \text{tap2tap}] + 2)\}$$

Figure 48-43. Equation for SCL Divider

The SDA Hold delay is equal to the CPU clock period multiplied by the SDA Hold value shown in Table 48-51. The equation used to generate the SDA Hold value from the IBFD bits is:

$$\text{SDA Hold} = \text{MUL} \times \{\text{scl2tap} + [(\text{SDA\_Tap} - 1) \times \text{tap2tap}] + 3\}$$

Figure 48-44. Equation for SDA Hold

The equations for SCL Hold values to generate the START and STOP conditions from the IBFD bits are:

$$\text{SCL Hold (start)} = \text{MUL} \times [\text{scl2start} + (\text{SCL\_Tap} - 1) \times \text{tap2tap}]$$

Figure 48-45. Equation for SCL Hold(start)

$$\text{SCL Hold (end)} = \text{MUL} \times [\text{scl2end} + (\text{SCL\_Tap} - 1) \times \text{tap2tap}]$$

Figure 48-46. Equation for SCL Hold(stop)

Table 48-51. I^{2}C divider and hold values

	IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
MUL=1	00	20	7	6	11
	01	22	7	7	12
	02	24	8	8	13
	03	26	8	9	14
	04	28	9	10	15
	05	30	9	11	16
	06	34	10	13	18
	07	40	10	16	21
	08	28	7	10	15
	09	32	7	12	17
	0A	36	9	14	19
	0B	40	9	16	21
	0C	44	11	18	23
	0D	48	11	20	25
	0E	56	13	24	29
	0F	68	13	30	35
	10	48	9	18	25
	11	56	9	22	29
	12	64	13	26	33
	13	72	13	30	37
	14	80	17	34	41
	15	88	17	38	45
	16	104	21	46	53
	17	128	21	58	65
	18	80	9	38	41
	19	96	9	46	49
	1A	112	17	54	57
	1B	128	17	62	65
	1C	144	25	70	73
	1D	160	25	78	81
	1E	192	33	94	97
	1F	240	33	118	121
	20	160	17	78	81
	21	192	17	94	97
	22	224	33	110	113
	23	256	33	126	129

Table continues on the next page...

Table 48-51. I^2C divider and hold values (continued)

	IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
MUL=1	24	288	49	142	145
	25	320	49	158	161
	26	384	65	190	193
	27	480	65	238	241
	28	320	33	158	161
	29	384	33	190	193
	2A	448	65	222	225
	2B	512	65	254	257
	2C	576	97	286	289
	2D	640	97	318	321
	2E	768	129	382	385
	2F	960	129	478	481
	30	640	65	318	321
	31	768	65	382	385
	32	896	129	446	449
	33	1024	129	510	513
	34	1152	193	574	577
	35	1280	193	638	641
	36	1536	257	766	769
	37	1920	257	958	961
	38	1280	129	638	641
	39	1536	129	766	769
	3A	1792	257	894	897
	3B	2048	257	1022	1025
	3C	2304	385	1150	1153
	3D	2560	385	1278	1281
	3E	3072	513	1534	1537
	3F	3840	513	1918	1921
MUL=2	40	40	14	12	22
	41	44	14	14	24
	42	48	16	16	26
	43	52	16	18	28
	44	56	18	20	30
	45	60	18	22	32
	46	68	20	26	36
	47	80	20	32	42
	48	56	14	20	30
	49	64	14	24	34

Table continues on the next page...

Table 48-51. I2C divider and hold values (continued)

	IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
MUL=2	4A	72	18	28	38
	4B	80	18	32	42
	4C	88	22	36	46
	4D	96	22	40	50
	4E	112	26	48	58
	4F	136	26	60	70
	50	96	18	36	50
	51	112	18	44	58
	52	128	26	52	66
	53	144	26	60	74
	54	160	34	68	82
	55	176	34	76	90
	56	208	42	92	106
	57	256	42	116	130
	58	160	18	76	82
	59	192	18	92	98
	5A	224	34	108	114
	5B	256	34	124	130
	5C	288	50	140	146
	5D	320	50	156	162
	5E	384	66	188	194
	5F	480	66	236	242
	60	320	34	156	162
	61	384	34	188	194
	62	448	66	220	226
	63	512	66	252	258
	64	576	98	284	290
	65	640	98	316	322
	66	768	130	380	386
	67	960	130	476	482
	68	640	66	316	322
	69	768	66	380	386
	6A	896	130	444	450
	6B	1024	130	508	514

Table continues on the next page...

Table 48-51. I^2C divider and hold values (continued)

	IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
MUL=2	6C	1152	194	572	578
	6D	1280	194	636	642
	6E	1536	258	764	770
	6F	1920	258	956	96
	70	1280	130	636	642
	71	1536	130	764	770
	72	1792	258	892	898
	73	2048	258	1020	1026
	74	2304	386	1148	1154
	75	2560	386	1276	1282
	76	3072	514	1532	1538
	77	3840	514	1916	1922
	78	2560	258	1276	1282
	79	3072	258	1532	1538
	7A	3584	514	1788	1794
	7B	4096	514	2044	2050
	7C	4608	770	2300	2306
	7D	5120	770	2556	2562
	7E	6144	1026	3068	3074
	7F	7680	1026	3836	3842

Table continues on the next page...

Table 48-51. I2C divider and hold values (continued)

	IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
MUL=4	80	80	28	24	44
	81	88	28	28	48
	82	96	32	32	52
	83	104	32	36	56
	84	112	36	40	60
	85	120	36	44	64
	86	136	40	52	72
	87	160	40	64	84
	88	112	28	40	60
	89	128	28	48	68
	8A	144	36	56	76
	8B	160	36	64	84
	8C	176	44	72	92
	8D	192	44	80	100
	8E	224	52	96	116
	8F	272	52	120	140
	90	192	36	72	100
	91	224	36	88	116
	92	256	52	104	132

Table continues on the next page...

Table 48-51. I2C divider and hold values (continued)

	IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
MUL=4	93	288	52	120	148
	94	320	68	136	164
	95	352	68	152	180
	96	416	84	184	212
	97	512	84	232	260
	98	320	36	152	164
	99	384	36	184	196
	9A	448	68	216	228
	9B	512	68	248	260
	9C	576	100	280	292
	9D	640	100	312	324
	9E	768	132	376	388
	9F	960	132	472	484
	A0	640	68	312	324
	A1	768	68	376	388
	A2	896	132	440	452
	A3	1024	132	504	516
	A4	1152	196	568	580
	A5	1280	196	632	644
	A6	1536	260	760	772
	A7	1920	260	952	964
	A8	1280	132	632	644
	A9	1536	132	760	772
	AA	1792	260	888	900
	AB	2048	260	1016	1028
	AC	2304	388	1144	1156
	AD	2560	388	1272	1284
	AE	3072	516	1528	1540
	AF	3840	516	1912	1924
	30	2560	260	1272	1284
	B1	3072	260	1528	1540
	B2	3584	516	1784	1796
	B3	4096	516	2040	2052
	B4	4608	772	2296	2308
	B5	5120	772	2552	2564
	B6	6144	1028	3064	3076
	B7	7680	1028	3832	3844
	B8	5120	516	2552	2564

Table continues on the next page...

Table 48-51. I²C divider and hold values (continued)

	IBC (hex)	SCL Divider (clocks)	SDA Hold (clocks)	SCL Hold (start)	SCL Hold (stop)
MUL=4	B9	6144	516	3064	3076
	BA	7168	1028	3576	3588
	BB	8192	1028	4088	4100
	BC	9216	1540	4600	4612
	BD	10240	1540	5112	5124
	BE	12288	2052	6136	6148
	BF	15360	2052	7672	7684

48.5.5 Interrupts

This section presents the following topics:

- Interrupt vector
- Interrupt description

48.5.5.1 Interrupt vector

The I²C module uses only one interrupt vector.

Table 48-52. Interrupt summary

Interrupt	Offset	Vector	Priority	Source	Description
I²C Interrupt	—	—	—	IBAL, TCF, IAAS, IBB bits in IBSR register	When any of IBAL, TCF or IAAS bits is set, an interrupt may be caused based on Arbitration lost, Transfer Complete or Address Detect conditions. If enabled by BIIE, the de-assertion of IBB can also cause an interrupt, indicating that the bus is idle.

48.5.5.2 Interrupt description

There are five types of internal interrupts in the I²C. The interrupt service routine can determine the interrupt type by reading the Status register.

I²C Interrupt can be generated on the following events:

- Arbitration Lost condition (IBAL bit set)
- Byte Transfer condition (TCF bit set and DMAEN bit not set)
- Address Detect condition (IAAS bit set)
- No Acknowledge from slave received when expected
- Bus Going Idle (IBB bit not set)

The I^2C interrupt is enabled by the IBCR[IBIE] bit. It must be cleared by writing '1' to the IBIF bit in the interrupt service routine. The Bus Going Idle interrupt needs to be additionally enabled by the IBIC[BIIE] bit.

48.5.6 IPG DEBUG mode

This mode allows CPU to debug the I^2C by freezing all the counters, registers and status bits. Once the Debug request is asserted along with IBDBG[IPG_DEBUG_EN] bit, I^2C comes to a graceful halt after completing all the ongoing transactions. A Debug halted signal IBDBG[IPG_DEBUG_HALTED] is also asserted to indicate that the debug request has been successfully serviced and is de-asserted when the Debug request is de-asserted.

As soon as the I^2C module enters IPG DEBUG mode:

- No transaction can take place.
- All the registers, counters and status bits are frozen. They all can be accessed by the CPU to enable the debugging.

The I^2C module enters this mode only after successfully completing the current ongoing transaction. For example, if the current transaction consists of 8 bytes and the debug mode was initiated by user at the time of the second byte, the IPG Debug Halted signal will be asserted after the 8th byte is transmitted/received. See the simulation result in Figure 48-47 for more details.

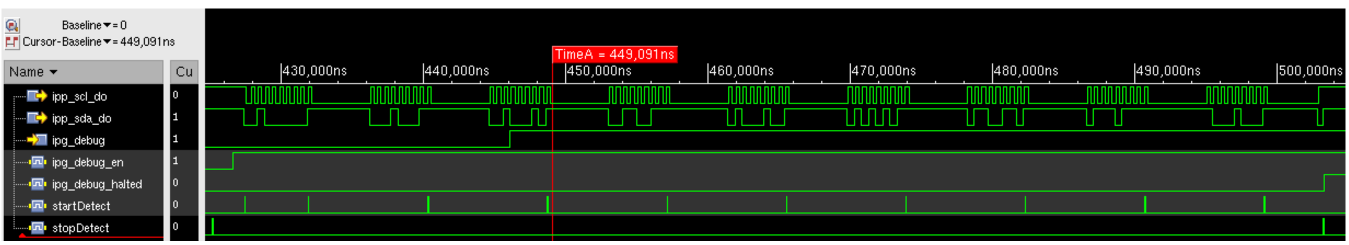


Figure 48-47. Simulation result of IPG Debug Halted in case of frame transmission

If any DMA transaction (transmit or receive) is in progress, and if the debug mode is requested while a byte is being transmitted or received, the I²C module enters IPG DEBUG mode after completing the transmission or reception of the current byte. As soon as the module exits IPG DEBUG mode, transmission or reception of the remaining bytes resumes. Please see the simulation snapshot shown in Figure 48-48 for details where the I²C is transmitting 8 bytes using DMA and IPG DEBUG mode is requested while a second byte is being transmitted. IPG DEBUG mode is entered after successfully transmitting the second byte (IPG Debug Halted signal asserted). As soon as the module exits IPG DEBUG mode (IPG Debug Halted signal de-asserted), transmission resumes successfully.

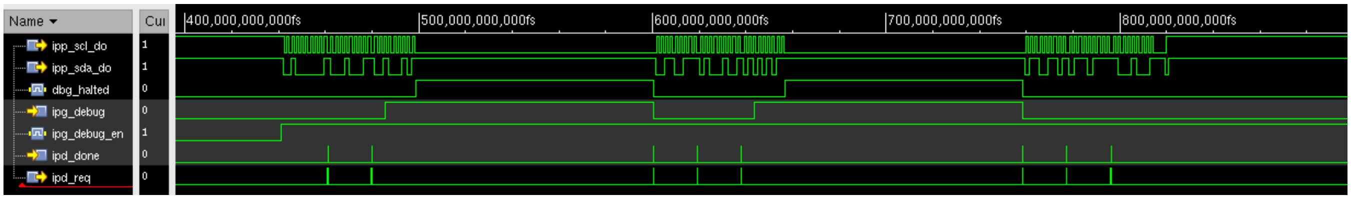


Figure 48-48. Simulation result of IPG Debug Halted in case of frame transmission using DMA

No more transaction can take place until the debug signal is de-asserted after which the I²C module starts functioning normally. There is a status halted signal IBDBG[IPG_DEBUG_HALTED] to indicate to the user that the I²C module has entered the IPG DEBUG mode. See the following figure for more details.

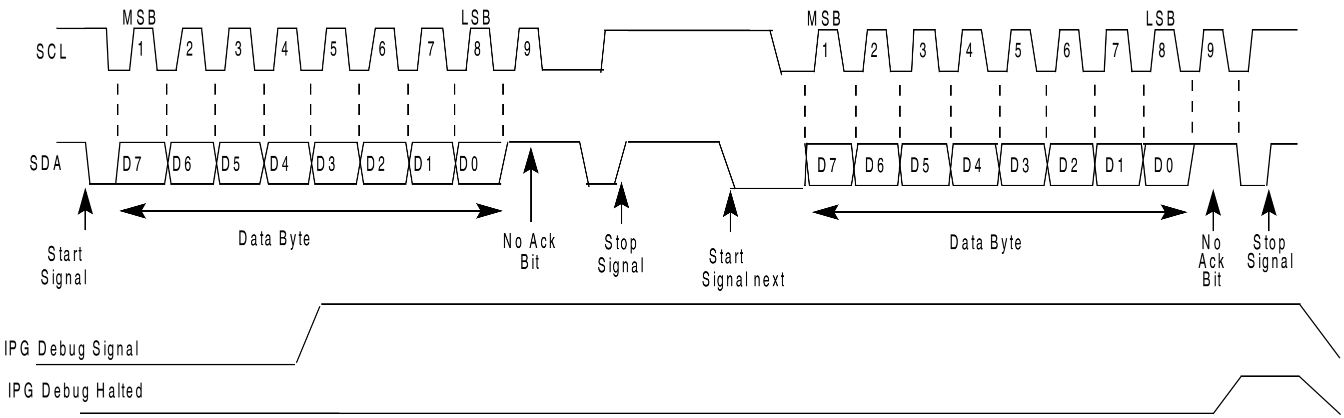


Figure 48-49. IPG DEBUG mode

The following figure shows a case of IPG DEBUG mode with repeated START transaction. In this case, if the debug signal is asserted in between the transaction, the entire transaction with multiple repeated start will be completed before the I²C module enters IPG DEBUG mode.

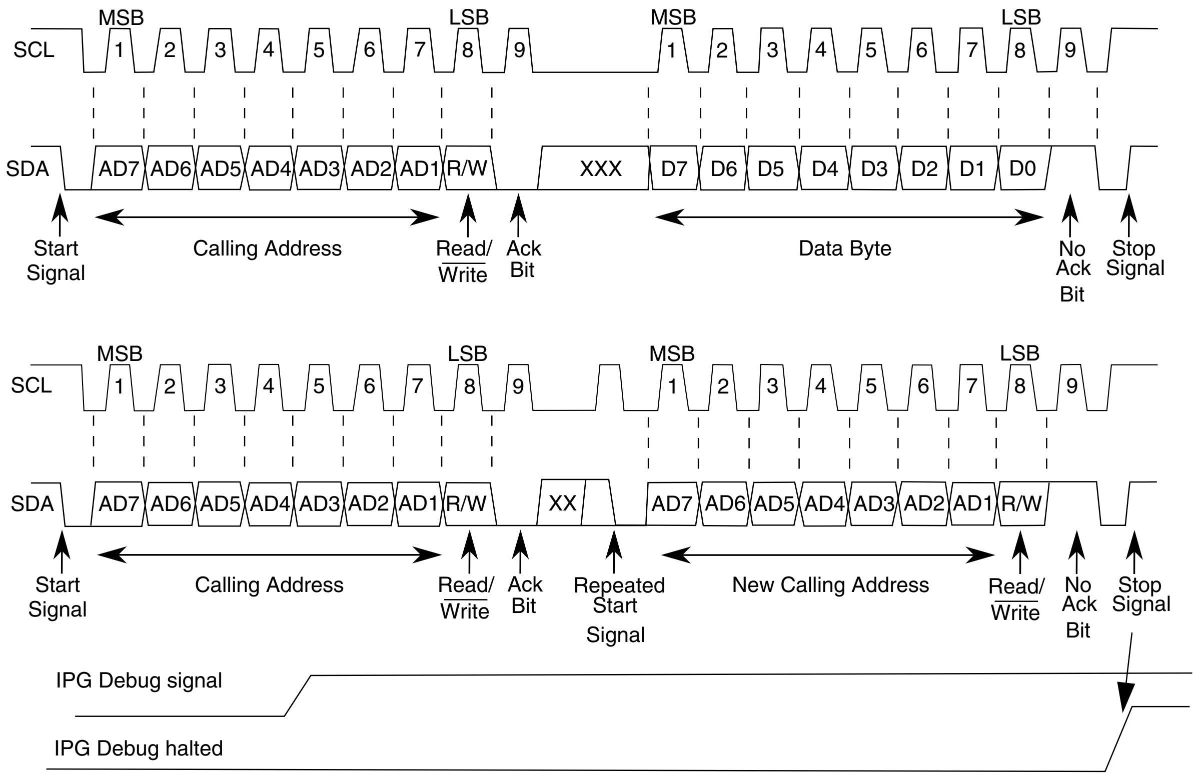


Figure 48-50. IPG debug mode with repeated start

48.5.7 DMA interface

A simple DMA interface is implemented so that the I^2C can request data transfers with minimal support from the CPU (see DMA application information). DMA mode is enabled by setting bit 1 in the Control Register (IBCR).

The DMA interface is operational when the I^2C module is configured for Master mode.

At least three bytes of data per frame must be transferred from/to the slave when using DMA mode, although in practice it will only be worthwhile using the DMA mode when there is a large number of data bytes to transfer per frame.

Two internal signals, TX request and RX request, are used to signal the DMA controller when the I^2C module requires data to be written or read from the data register.

Further details of the DMA interface can be found in the Initialization/application information, of this document.

48.6 Initialization/application information

This section presents the following topics:

- Recommended interrupt service flow
- General programming guidelines (for both master and slave mode)
- Programming guidelines specific to master mode
- Programming guidelines specific to slave mode
- DMA application information

48.6.1 Recommended interrupt service flow

The following figure shows a flowchart for the recommended I^2C interrupt service routine. Deviation from the flowchart may result in unpredictable I^2C bus behavior.

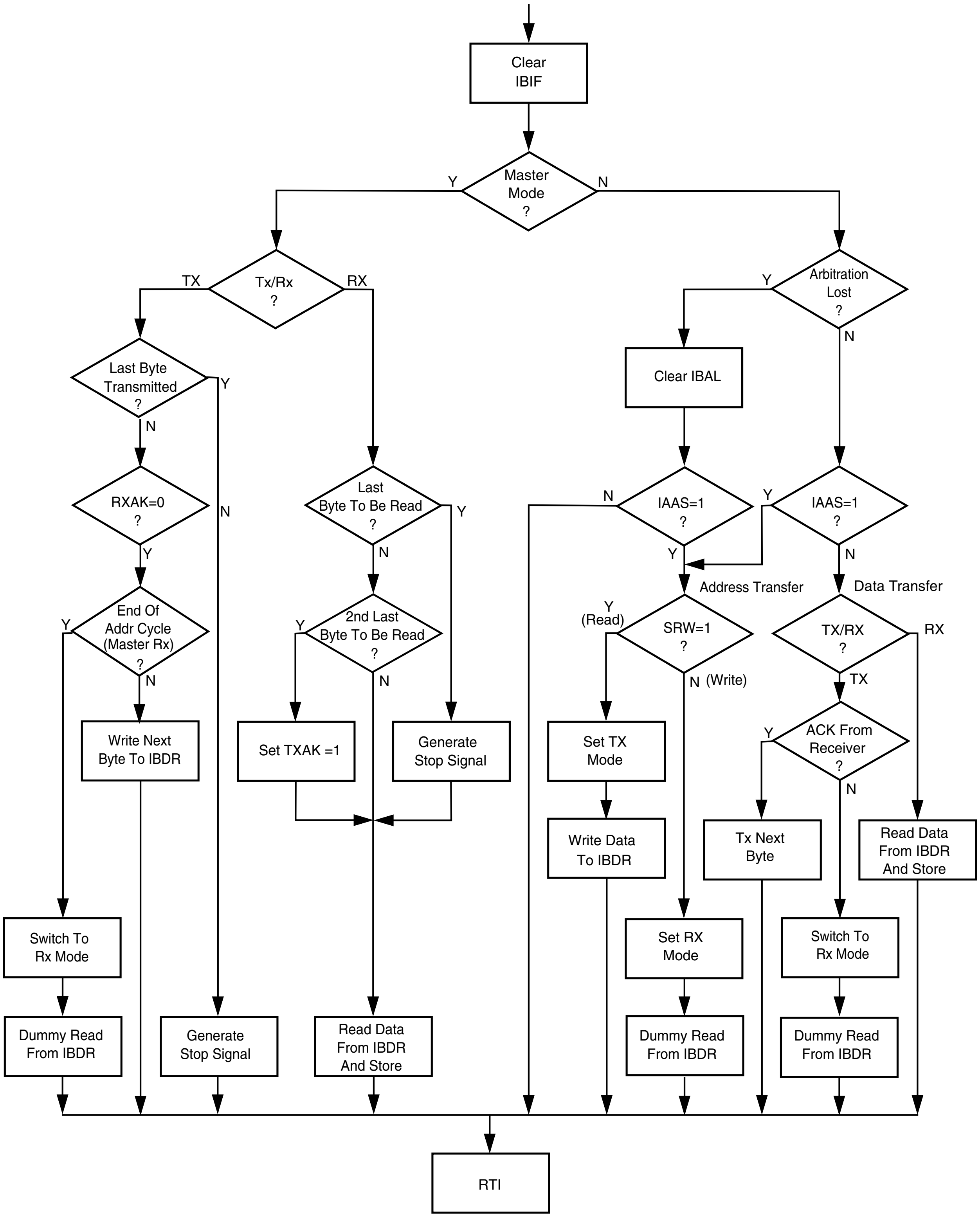


Figure 48-51. Recommended I^2C interrupt service routine flowchart

48.6.2 General programming guidelines (for both master and slave mode)

This section provides programming guidelines recommended for the I^{2}C module in both master and slave mode. It presents the following topics:

- Section 1.6.2.1, Initializing the I^{2}C module
- Software response after a transfer

48.6.2.1 Initializing the I^{2}C module

The following sequence initializes the I^{2}C module:

1. Use the IBFD register to select the required division ratio to obtain SCL frequency from system clock.
2. Use the IBAD register to define the slave address.
3. Clear the IBCR[MDIS] bit to enable the I^{2}C interface system.
4. Use the IBCR to select Master/Slave mode, Transmit/Receive mode and interrupt enable or not.
5. (Optional) Use the IBIC register to further refine the interrupt behavior.

48.6.2.2 Software response after a transfer

Transmission or reception of a byte will set the data transferring bit (TCF) to 1, which indicates one byte communication is finished. The I^{2}C Bus interrupt bit (IBIF) is set also; an interrupt will be generated if the interrupt function is enabled during initialization by setting the IBIE bit. The IBIF (interrupt flag) can be cleared by writing one (in the interrupt service routine, if interrupts are used).

The TCF bit will be cleared to indicate data transfer in progress whenever data register is written to in transmit mode, or during reading out from data register in receive mode. The TCF bit should not be used as a data transfer complete flag as the flag timing is dependent on a number of factors including the I^{2}C bus frequency. This bit may not conclusively provide an indication of a transfer complete situation. It is recommended that transfer complete situations are detected using the IBIF flag.

Software may service the I^2C I/O in the main program by monitoring the IBIF bit if the interrupt function is disabled. Polling should monitor the IBIF bit rather than the TCF bit since their operation is different when arbitration is lost.

When a "Transfer Complete" interrupt occurs at the end of the address cycle, the master will always be in transmit mode, i.e. the address is transmitted. If master receive mode is required, indicated by R/W bit sent with slave calling address, then the Tx/Rx bit at Master side should be toggled at this stage. If Master does not receive an ACK from Slave, then transmission must be re-initiated or terminated.

In slave mode, IAAS bit will get set in IBSR if Slave address (IBAD) matches the Master calling address. This is an indication that Master-Slave data communication can now start. During address cycles (IBSR[IAAS]=1), the SRW bit in the status register is read to determine the direction of the subsequent transfer and the Tx/Rx bit is programmed accordingly. For slave mode data cycles (IBSR[IAAS]=0), the SRW bit is not valid. The Tx/Rx bit in the control register should be read to determine the direction of the current transfer.

48.6.3 Programming guidelines specific to master mode

This section presents the following topics:

- Generating START
- Transmit/receive sequence
- Generating STOP
- Generating repeated START
- Loss of arbitration

48.6.3.1 Generating START

After completion of the initialization procedure, serial data can be transmitted by selecting the 'master transmitter' mode. If the device is connected to a multi-master bus system, the state of the I^2C Bus Busy bit (IBB) must be tested to check whether the serial bus is free.

If the bus is free (IBB=0), the start condition and the first byte (the slave address) can be sent. The data written to the data register comprises the slave calling address and the LSB, which is set to indicate the direction of transfer required from the slave.

The bus free time (i.e., the time between a STOP condition and the following START condition) is built into the hardware that generates the START cycle. Depending on the relative frequencies of the system clock and the SCL period, it may be necessary to wait until the I^2C is busy after writing the calling address to the IBDR before proceeding with the following instructions. This is illustrated in the following example.

An example of the sequence of events which generates the START signal and transmits the first byte of data (slave address) is shown below:

```
while (IBSR[IBB]==1)// wait in loop for IBB flag to clear
IBCR[MS/MSL] and IBVR[]Tx/Rx] = 1//master and transmit mode, that is, generate start
condition
IBDR = calling_address// send the calling address to the data register
while (bit 5, IBSR ==0)// wait in loop for IBB flag to be set
```

48.6.3.2 Transmit/receive sequence

Below sequence to be followed in case of Master Transmit (Address/Data):

1. Clear IBIF in IBSR register.
2. Write data in Data Register (IBDR).
3. TCF bit will get cleared in IBSR register when transfer is in progress.
4. TCF bit will get set in IBSR register when transfer is complete.
5. Wait for IBIF to get set in IBSR register, then read IBSR register to determine its source
    - TCF = 1 that is transfer is complete.
    - No Acknowledge condition (RXAK = 1) is found.
    - IBB = 0 that is Bus has transitioned from Busy to Idle state.
    - If IBB = 1, ignore check of Arbitration Loss (IBAL = 1).
    - Ignore Address Detect (IAAS = 1) for Master mode (valid only for Slave mode).
6. Check RXAK in IBSR for an acknowledge from slave.

Below sequence to be followed in case of Slave Receive (Address/Data):

1. Clear IBIF in IBSR register.
2. TCF bit will get cleared in IBSR register when transfer is in progress for address transfer.

3. TCF bit will get set in IBSR register when transfer is complete.
4. Wait for IBIF to get set in IBSR register. Then read IBSR register to determine its source:
   - Address Detect has occurred (IAAS = 1) — determination of Slave mode.
5. Clear IBIF.
6. Wait till TCF bit gets cleared in IBSR register (that is "Transfer under Progress" condition is reached for data transfer).
7. Wait till TCF bit gets cleared in IBSR register (proof that Transfer Completes from "Transfer under Progress" state).
8. Wait till IBIF bit gets set in IBSR register. To find its source, check if:
   - TCF = 1 that is reception is complete.
   - IBB= 0 in IBSR register i.e. Bus has transitioned from Busy to Idle state.
   - Ignore Arbitration Loss (IBAL = 1) for IBB = 1.
   - Ignore No Acknowledge condition (RXAK = 1) for receiver.
9. Read the Data Register (IBDR) to determine data received from Master.

Sequence followed in case of Slave Transmit (Steps a-d of Slave Receive for Address Detect, followed by a-f of Master Transmit for Data Transmit).

Sequence followed in case of Master Receive (Steps a-f of Master Transmit for Address dispatch, followed by e-h of Slave Receive for Data Receive)

48.6.3.3 Generating STOP

A data transfer ends with a STOP signal generated by the 'master' device. A master transmitter can simply generate a STOP signal after all the data has been transmitted. The following is an example showing how a STOP condition is generated by a master transmitter.

```
if (tx_count == 0) or// check to see if all data bytes have been transmitted
   (bit 0, IBSR == 1) {// or if no ACK generated
     clear bit 5, IBCR// generate stop condition
   }
else {
  IBDR = data_to_transmit// write byte of data to DATA register
    tx_count --// decrement counter
  } // return from interrupt
```

If a master receiver wants to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last byte of data which can be done by setting the NOACK bit in IBCR before reading the 2nd last byte of data. Before reading the last byte of data, a STOP signal must first be generated. The following is an example showing how a STOP signal is generated by a master receiver.

```
rx_count --// decrease the rx counter
if (rx_count ==1)// 2nd last byte to be read ?
 bit 3, IBCR = 1// disable ACK
 if (rx_count == 0)// last byte to be read ?
 bit 5, IBCR = 0// generate stop signal
else
data_received = IBDR// read RX data and store
```

48.6.3.4 Generating repeated START

At the end of data transfer, if the master still wants to communicate on the bus, it can generate another START signal followed by another slave address without first generating a STOP signal. A program example is as shown.

```
bit 2, IBCR = 1                                    // generate another start (restart)
IBDR == calling_address                        // transmit the calling address
```

48.6.3.5 Loss of arbitration

If several masters try to engage the bus simultaneously, only one master wins and the others lose arbitration. The devices that lost arbitration are immediately switched to slave receive mode by the hardware. Their data output to the SDA line is stopped, but SCL is still generated until the end of the byte during which arbitration was lost. An interrupt occurs at the falling edge of the ninth clock of this transfer with IBAL=1 and MS/SL=0. If one master attempts to start transmission, while the bus is being engaged by another master, the hardware will inhibit the transmission, switch the MS/SL bit from 1 to 0 without generating a STOP condition, generate an interrupt to CPU and set the IBAL to indicate that the attempt to engage the bus is failed. When considering these cases, the slave service routine should test the IBAL first and the software should clear the IBAL bit if it is set.

48.6.4 Programming guidelines specific to slave mode

In the slave interrupt service routine, the module addressed as slave bit (IAAS) should be tested to check if a calling of its own address has just been received. If IAAS is set, software should set the transmit/receive mode select bit (Tx/Rx bit of IBCR) according to

the R/W command bit (SRW). Writing to the IBCR clears IAAS automatically. Note that the only time IAAS is read as set is from the interrupt at the end of the address cycle where an address match occurred. Interrupts resulting from subsequent data transfers will have IAAS cleared. A data transfer may now be initiated by writing information to IBDR for slave transmits or dummy reading from IBDR in slave receive mode. The slave will drive SCL low in-between byte transfers, SCL is released when the IBDR is accessed in the required mode.

In slave transmitter routine, the received acknowledge bit (RXAK) must be tested before transmitting the next byte of data. Setting RXAK means an "end of data" signal from the master receiver, after which it must be switched from transmitter mode to receiver mode by software. A dummy read then releases the SCL line so that the master can generate a STOP signal.

48.6.5 DMA application information

The DMA interface on the I^2C is not completely autonomous and requires intervention from the CPU to start and to terminate the frame transfer. DMA mode is only valid for Master transmit and Master receive modes. Software must ensure that the DMA enable bit in the control register is not set when the I^2C module is configured in slave mode.

The DMA controller must only transfer one byte of data per Tx/Rx request. This is because there is no FIFO on the I^2C block.

The CPU should also keep the I^2C interrupt enabled during a DMA transfer to detect the arbitration lost condition and take action to recover from this situation. The DMAEN bit in the IBCR register works as a disable for the transfer complete interrupt. This means that during normal transfers (no errors) there will always be either an interrupt or a request to the DMA controller, dependant on the setting of the DMAEN bit. All error conditions will trigger an interrupt and require CPU intervention. The address match condition will not occur in DMA mode as the I^2C should never be configured for slave operation.

The following sections detail how to set up a DMA transfer and what intervention is required from the CPU. It is assumed that the system DMA controller is capable of generating an interrupt after a certain number of DMA transfers have taken place. The sections present the following topics:

- DMA mode, master transmit
- DMA mode, master reception
- Exiting DMA mode, system requirement considerations

48.6.5.1 DMA mode, master transmit

The following flow diagram details exactly the operation for using a DMA controller to transmit "n" data bytes to a slave. The first byte (the slave calling address) is always transmitted by the CPU. All subsequent data bytes (apart from the last data byte) can be transferred by the DMA controller. The last data byte must be transferred by the CPU.

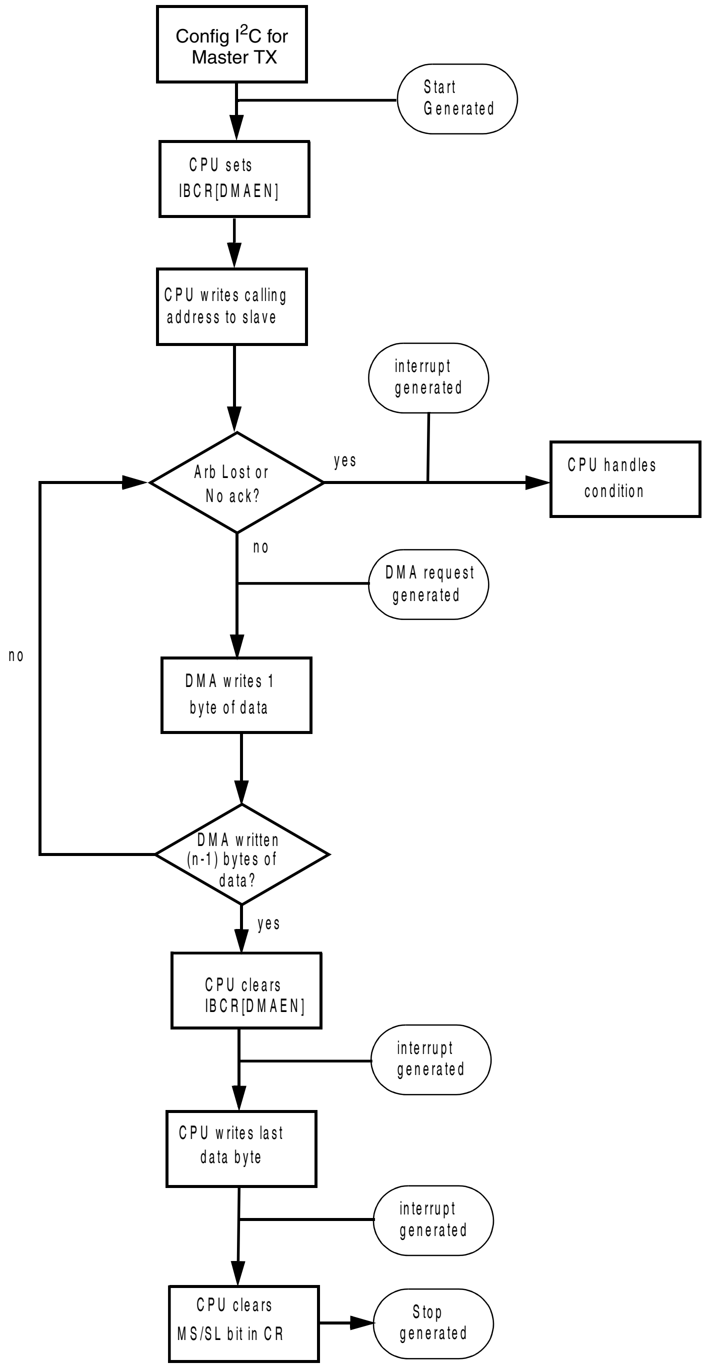


Figure 48-52. Flow-Chart of DMA mode master transmit

48.6.5.2 DMA mode, master reception

The following flow diagram details the exact operation for using a DMA controller to receive "n" data bytes from a slave. The first byte (the slave calling address) is always transmitted by the CPU. All subsequent data bytes (apart from the two last data bytes) can be read by the DMA controller. The last two data bytes must be transferred by the CPU.

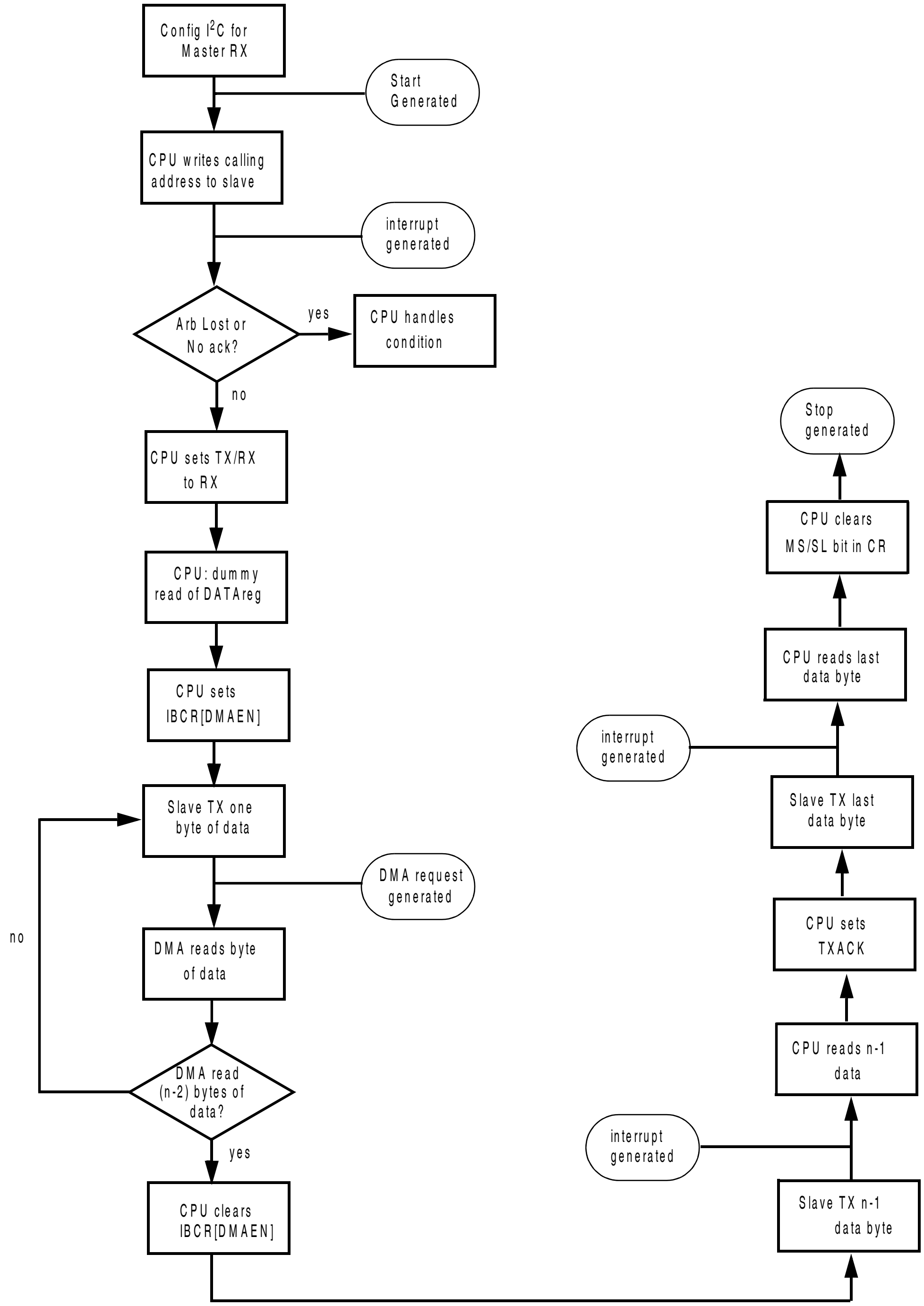


Figure 48-53. Flow-Chart of DMA mode master receive

48.6.5.3 Exiting DMA mode, system requirement considerations

As described above, the final transfers of both TX and RX transfers need to be handled via interrupt by the CPU. To change from DMA to interrupt driven transfers in the I^2C module, you have to disable the DMAEN bit in the IBCR register. The trigger to exit the DMA mode is that the programmed DMA Transfer Control Descriptor (TCD) has completed all its transfers to/from the I^2C module.

After the last DMA write (TX mode) to the I^2C the module will immediately start the next I^2C-bus transfer. The same is true for receive mode. After the DMA read from the IBDR register the module initiates the next I^2C-bus transfer. This results in two possible scenarios in the DMA mode exiting scheme.

1. Fast reaction

   The DMAEN bit is cleared before the next I^2C-bus transfer completes. In this case the module will raise an interrupt request to the CPU which can be serviced normally.

2. Slow reaction

   The DMAEN bit is cleared after the next I^2C-bus transfer has already completed. In this case, the module will not raise an interrupt request to the CPU. Instead the TCF bit can be read to determine that the transfer completed and the module is ready for further transfer.

What is fast/slow reaction?

The reaction time T_R for the system to disable DMAEN after the last DMA controller access to the I^2C is the time required for one byte transfer over the I^2C. For 'fast reaction' the disabling has to occur before the 9^{th} bit of the data transfer which is the ACK bit. So the time available is eight times the SCL period.

$$T_R = 8 \times T_{SCL}$$

In fast mode, with 400kbit/s, T_{SCL} is 2.5μs, so T_R is 20μs.

Depending on the system and DMA controller there are different possibilities for the de-assertion of DMAEN. Three options are:

1. CPU intervention via Interrupt

   The DMA controller is programmed to signal an interrupt to the CPU which is then responsible for the de-assertion of DMAEN. This scheme should be supported by most systems but it can result in a slow reaction time if other higher priority interrupts interfere. Therefore the interrupt handling routine can become complicated as it has to check which of the two cases happened (check TCF bit) and act

accordingly. In case of slow reaction you can force an interrupt for the I^2C in the interrupt controller to have the further transfer handled by the normal I^2C interrupt routine.

Note

The use of nested interrupts can still cause potential issues in this scenario, if someone tries to stall the DMA interrupt between the de-assertion and DMAEN bit and checks the TCF bit.

2. DMA channel linking (if supported)

   The Transfer control descriptor in the DMA controller that performs the data transfer is linked to another channel that does a write to the I^2C IBCR register to disable the DMAEN bit. This is probably the fastest system solution, but it uses two DMA channels.

Note

Here you have to make sure on system level that no higher priority DMA requests occur between the two linked TCDs as those could again create a scenario of slow reaction.

3. DMA scatter/gather process (if supported)

   The Transfer control descriptor in the DMA controller that performs the data transfer has the scatter/gather feature activated. This feature will initiate a reload of another TCD from system RAM after the completion of the first TCD. The new TCD will have its start bit already set and immediately start the required write to the I^2C IBCR register to disable the DMAEN bit. This TCD also has scatter/gather activated and is programmed to reload the initial TCD upon completion, bringing the system back into a "ready-for-I^2C-transfer" state. The advantage over the two other solutions is that this neither requires CPU intervention nor a 2^{nd} DMA channel. This comes at the cost of 64 bytes RAM (two TCDs), some system bus transfer overhead and a little increase in application code complexity.

Note

Here you have to make sure at system level that no higher priority DMA requests occur during the scatter/gather process, as those could again create a scenario of slow reaction.

Example latencies for a 32MHz system with a full speed 32-bit AHB bus and an I^2C connected via half speed IPI bus:

- Accessing the I^2C from the DMA controller via IPI bus typically requires four cycles (consecutive accesses to the I^2C could be faster):

  $$4 \times T_{IPI} = 4/16\,MHz = 250\,ns$$

- Reloading a new TCD (8 x 32-bit) via AHB to the DMA controller (scatter/gather process):

  $$8 \times T_{AHD} = 8/32\,MHz = 250\,ns$$

With the DMA scatter/gather process the required IBCR access can be done in 0.5 µs, leaving a large margin of 19.5 µs for additional system delays. In this way, the slow reaction case can be prevented. The system user needs to decide which usage model best suits his overall requirement.

Chapter 49
Universal Asynchronous Receiver/Transmitter (UART)

49.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The UART allows asynchronous serial communication with peripheral devices and CPUs.

49.1.1 Features

The UART includes the following features:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse width
- 13-bit baud rate selection with /32 fractional divide, based on the module clock frequency
- Programmable 8-bit or 9-bit data format
- Separately enabled transmitter and receiver
- Programmable transmitter output polarity
- Programmable receive input polarity
- Up to 14-bit break character transmission.

- 11-bit break character detection option
- Independent FIFO structure for transmit and receive
- Two receiver wakeup methods:
    - Idle line wakeup
    - Address mark wakeup
- Address match feature in the receiver to reduce address mark wakeup ISR overhead
- Ability to select MSB or LSB to be first bit on wire
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Support for ISO 7816 protocol to interface with SIM cards and smart cards
    - Support for T=0 and T=1 protocols
    - Automatic retransmission of NACK'd packets with programmable retry threshold
    - Support for 11 and 12 ETU transfers
    - Detection of initial packet and automated transfer parameter programming
    - Interrupt-driven operation with seven ISO-7816 specific interrupts:
        - Wait time violated
        - Character wait time violated
        - Block wait time violated
        - Initial frame detected
        - Transmit error threshold exceeded
        - Receive error threshold exceeded
        - Guard time violated
- Support for CEA709.1-B protocol used in building automation and home networking systems
    - Automatic clock resynchronization
    - Support for collision detection
- Interrupt-driven operation with 12 flags, not specific to ISO-7816 support

  - Transmitter data buffer at or below watermark
  - Transmission complete
  - Receiver data buffer at or above watermark
  - Idle receiver input
  - Receiver data buffer overrun
  - Receiver data buffer underflow
  - Transmit data buffer overflow
  - Noise error
  - Framing error
  - Parity error
  - Active edge on receive pin
  - LIN break detect
- Receiver framing error detection
- Hardware parity generation and checking
- 1/16 bit-time noise detection
- DMA interface

49.1.2 Modes of operation

The UART functions in the same way in all the normal modes.

It has the following low power mode:

- Stop mode

49.1.2.1 Run mode

This is the normal mode of operation.

49.1.2.2 Stop mode

The UART is inactive during Stop mode for reduced power consumption. The STOP instruction does not affect the UART register states, but the UART module clock is disabled. The UART operation resumes after an external interrupt brings the CPU out of Stop mode. Bringing the CPU out of Stop mode by reset aborts any ongoing transmission or reception and resets the UART. Entering or leaving Stop mode does not initiate any power down or power up procedures for the ISO-7816 smartcard interface.

49.2 UART signal descriptions

The UART signals are shown in the following table.

Table 49-1. UART signal descriptions

Signal	Description	I/O
CTS	Clear to send	I
$\overline{\text{RTS}}$	Request to send	O
RXD	Receive data	I
TXD	Transmit data	O
Collision	Collision detect	I

49.2.1 Detailed signal descriptions

The detailed signal descriptions of the UART are shown in the following table.

Table 49-2. UART—Detailed signal descriptions

<table>
<tr><th>Signal</th><th>I/O</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">CTS</td><td rowspan="3">I</td><td colspan="2">Clear to send. Indicates whether the UART can start transmitting data when flow control is enabled.</td></tr>
<tr><td>State meaning</td><td>Asserted—Data transmission can start.
Negated—Data transmission cannot start.</td></tr>
<tr><td>Timing</td><td>Assertion—When transmitting device's RTS asserts.
Negation—When transmitting device's RTS deasserts.</td></tr>
</table>

Table continues on the next page...

Table 49-2. UART—Detailed signal descriptions (continued)

Signal	I/O	Description	
RTS	O	Request to send. When driven by the receiver, indicates whether the UART is ready to receive data. When driven by the transmitter, can enable an external transceiver during transmission.	
		State meaning	Asserted—When driven by the receiver, ready to receive data. When driven by the transmitter, enable the external transmitter. Negated—When driven by the receiver, not ready to receive data. When driven by the transmitter, disable the external transmitter.
		Timing	Assertion—Can occur at any time; can assert asynchronously to the other input signals. Negation—Can occur at any time; can deassert asynchronously to the other input signals.
RXD	I	Receive data. Serial data input to receiver.	
		State meaning	Whether RXD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
		Timing	Sampled at a frequency determined by the module clock divided by the baud rate.
TXD	O	Transmit data. Serial data output from transmitter.	
		State meaning	Whether TXD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
		Timing	Driven at the beginning or within a bit time according to the bit encoding method along with other configuration settings. Otherwise, transmissions are independent of reception timing.
Collision	I	Collision Detect. Indicates if a collision is detected during Data Transmission.	
		State meaning	Asserted—Indicates a collision detection. UARTxCPW determines the length of this pulse for valid collision detection. Negated—No collision detected.
		Timing	Asserts asynchronously to other input signals.

49.3 Memory map and registers

This section provides a detailed description of all memory and registers.

Accessing reserved addresses within the memory map may result in unpredicatable behavior, and the contents of implemented addresses may get modified as a result of that access.

Only byte accesses are supported.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_7000	UART Baud Rate Registers: High (UART0_BDH)	8	R/W	00h	49.3.1/2646
4002_7001	UART Baud Rate Registers: Low (UART0_BDL)	8	R/W	04h	49.3.2/2647
4002_7002	UART Control Register 1 (UART0_C1)	8	R/W	00h	49.3.3/2647
4002_7003	UART Control Register 2 (UART0_C2)	8	R/W	00h	49.3.4/2649
4002_7004	UART Status Register 1 (UART0_S1)	8	R	C0h	49.3.5/2651
4002_7005	UART Status Register 2 (UART0_S2)	8	R/W	00h	49.3.6/2654
4002_7006	UART Control Register 3 (UART0_C3)	8	R/W	00h	49.3.7/2656
4002_7007	UART Data Register (UART0_D)	8	R/W	00h	49.3.8/2657
4002_7008	UART Match Address Registers 1 (UART0_MA1)	8	R/W	00h	49.3.9/2658
4002_7009	UART Match Address Registers 2 (UART0_MA2)	8	R/W	00h	49.3.10/ 2659
4002_700A	UART Control Register 4 (UART0_C4)	8	R/W	00h	49.3.11/ 2659
4002_700B	UART Control Register 5 (UART0_C5)	8	R/W	00h	49.3.12/ 2660
4002_700C	UART Extended Data Register (UART0_ED)	8	R	00h	49.3.13/ 2661
4002_700D	UART Modem Register (UART0_MODEM)	8	R/W	00h	49.3.14/ 2662
4002_700E	UART Infrared Register (UART0_IR)	8	R/W	00h	49.3.15/ 2663
4002_7010	UART FIFO Parameters (UART0_PFIFO)	8	R/W	See section	49.3.16/ 2664
4002_7011	UART FIFO Control Register (UART0_CFIFO)	8	R/W	00h	49.3.17/ 2665
4002_7012	UART FIFO Status Register (UART0_SFIFO)	8	R/W	C0h	49.3.18/ 2666
4002_7013	UART FIFO Transmit Watermark (UART0_TWFIFO)	8	R/W	00h	49.3.19/ 2667
4002_7014	UART FIFO Transmit Count (UART0_TCFIFO)	8	R	00h	49.3.20/ 2668
4002_7015	UART FIFO Receive Watermark (UART0_RWFIFO)	8	R/W	01h	49.3.21/ 2668
4002_7016	UART FIFO Receive Count (UART0_RCFIFO)	8	R	00h	49.3.22/ 2669
4002_7018	UART 7816 Control Register (UART0_C7816)	8	R/W	00h	49.3.23/ 2669
4002_7019	UART 7816 Interrupt Enable Register (UART0_IE7816)	8	R/W	00h	49.3.24/ 2671
4002_701A	UART 7816 Interrupt Status Register (UART0_IS7816)	8	R/W	00h	49.3.25/ 2672
4002_701B	UART 7816 Wait Parameter Register (UART0_WP7816T0)	8	R/W	0Ah	49.3.26/ 2673

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_701B	UART 7816 Wait Parameter Register (UART0_WP7816T1)	8	R/W	0Ah	49.3.27/ 2674
4002_701C	UART 7816 Wait N Register (UART0_WN7816)	8	R/W	00h	49.3.28/ 2674
4002_701D	UART 7816 Wait FD Register (UART0_WF7816)	8	R/W	01h	49.3.29/ 2675
4002_701E	UART 7816 Error Threshold Register (UART0_ET7816)	8	R/W	00h	49.3.30/ 2675
4002_701F	UART 7816 Transmit Length Register (UART0_TL7816)	8	R/W	00h	49.3.31/ 2676
4002_7021	UART CEA709.1-B Control Register 6 (UART0_C6)	8	R/W	00h	49.3.32/ 2677
4002_7022	UART CEA709.1-B Packet Cycle Time Counter High (UART0_PCTH)	8	R/W	00h	49.3.33/ 2677
4002_7023	UART CEA709.1-B Packet Cycle Time Counter Low (UART0_PCTL)	8	R/W	00h	49.3.34/ 2678
4002_7024	UART CEA709.1-B Beta1 Timer (UART0_B1T)	8	R/W	00h	49.3.35/ 2678
4002_7025	UART CEA709.1-B Secondary Delay Timer High (UART0_SDTH)	8	R/W	00h	49.3.36/ 2679
4002_7026	UART CEA709.1-B Secondary Delay Timer Low (UART0_SDTL)	8	R/W	00h	49.3.37/ 2679
4002_7027	UART CEA709.1-B Preamble (UART0_PRE)	8	R/W	00h	49.3.38/ 2679
4002_7028	UART CEA709.1-B Transmit Packet Length (UART0_TPL)	8	R/W	00h	49.3.39/ 2680
4002_7029	UART CEA709.1-B Interrupt Enable Register (UART0_IE)	8	R/W	00h	49.3.40/ 2680
4002_702A	UART CEA709.1-B WBASE (UART0_WB)	8	R/W	00h	49.3.41/ 2681
4002_702B	UART CEA709.1-B Status Register (UART0_S3)	8	R/W	00h	49.3.42/ 2682
4002_702C	UART CEA709.1-B Status Register (UART0_S4)	8	R/W	00h	49.3.43/ 2683
4002_702D	UART CEA709.1-B Received Packet Length (UART0_RPL)	8	R	00h	49.3.44/ 2684
4002_702E	UART CEA709.1-B Received Preamble Length (UART0_RPREL)	8	R	00h	49.3.45/ 2685
4002_702F	UART CEA709.1-B Collision Pulse Width (UART0_CPW)	8	R/W	00h	49.3.46/ 2685
4002_7030	UART CEA709.1-B Receive Indeterminate Time (UART0_RIDT)	8	R/W	00h	49.3.47/ 2685
4002_7031	UART CEA709.1-B Transmit Indeterminate Time (UART0_TIDT)	8	R/W	00h	49.3.48/ 2686

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_8000	UART Baud Rate Registers: High (UART1_BDH)	8	R/W	00h	49.3.1/2646
4002_8001	UART Baud Rate Registers: Low (UART1_BDL)	8	R/W	04h	49.3.2/2647
4002_8002	UART Control Register 1 (UART1_C1)	8	R/W	00h	49.3.3/2647
4002_8003	UART Control Register 2 (UART1_C2)	8	R/W	00h	49.3.4/2649
4002_8004	UART Status Register 1 (UART1_S1)	8	R	C0h	49.3.5/2651
4002_8005	UART Status Register 2 (UART1_S2)	8	R/W	00h	49.3.6/2654
4002_8006	UART Control Register 3 (UART1_C3)	8	R/W	00h	49.3.7/2656
4002_8007	UART Data Register (UART1_D)	8	R/W	00h	49.3.8/2657
4002_8008	UART Match Address Registers 1 (UART1_MA1)	8	R/W	00h	49.3.9/2658
4002_8009	UART Match Address Registers 2 (UART1_MA2)	8	R/W	00h	49.3.10/ 2659
4002_800A	UART Control Register 4 (UART1_C4)	8	R/W	00h	49.3.11/ 2659
4002_800B	UART Control Register 5 (UART1_C5)	8	R/W	00h	49.3.12/ 2660
4002_800C	UART Extended Data Register (UART1_ED)	8	R	00h	49.3.13/ 2661
4002_800D	UART Modem Register (UART1_MODEM)	8	R/W	00h	49.3.14/ 2662
4002_800E	UART Infrared Register (UART1_IR)	8	R/W	00h	49.3.15/ 2663
4002_8010	UART FIFO Parameters (UART1_PFIFO)	8	R/W	See section	49.3.16/ 2664
4002_8011	UART FIFO Control Register (UART1_CFIFO)	8	R/W	00h	49.3.17/ 2665
4002_8012	UART FIFO Status Register (UART1_SFIFO)	8	R/W	C0h	49.3.18/ 2666
4002_8013	UART FIFO Transmit Watermark (UART1_TWFIFO)	8	R/W	00h	49.3.19/ 2667
4002_8014	UART FIFO Transmit Count (UART1_TCFIFO)	8	R	00h	49.3.20/ 2668
4002_8015	UART FIFO Receive Watermark (UART1_RWFIFO)	8	R/W	01h	49.3.21/ 2668
4002_8016	UART FIFO Receive Count (UART1_RCFIFO)	8	R	00h	49.3.22/ 2669
4002_8018	UART 7816 Control Register (UART1_C7816)	8	R/W	00h	49.3.23/ 2669
4002_8019	UART 7816 Interrupt Enable Register (UART1_IE7816)	8	R/W	00h	49.3.24/ 2671
4002_801A	UART 7816 Interrupt Status Register (UART1_IS7816)	8	R/W	00h	49.3.25/ 2672
4002_801B	UART 7816 Wait Parameter Register (UART1_WP7816T0)	8	R/W	0Ah	49.3.26/ 2673

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_801B	UART 7816 Wait Parameter Register (UART1_WP7816T1)	8	R/W	0Ah	49.3.27/ 2674
4002_801C	UART 7816 Wait N Register (UART1_WN7816)	8	R/W	00h	49.3.28/ 2674
4002_801D	UART 7816 Wait FD Register (UART1_WF7816)	8	R/W	01h	49.3.29/ 2675
4002_801E	UART 7816 Error Threshold Register (UART1_ET7816)	8	R/W	00h	49.3.30/ 2675
4002_801F	UART 7816 Transmit Length Register (UART1_TL7816)	8	R/W	00h	49.3.31/ 2676
4002_8021	UART CEA709.1-B Control Register 6 (UART1_C6)	8	R/W	00h	49.3.32/ 2677
4002_8022	UART CEA709.1-B Packet Cycle Time Counter High (UART1_PCTH)	8	R/W	00h	49.3.33/ 2677
4002_8023	UART CEA709.1-B Packet Cycle Time Counter Low (UART1_PCTL)	8	R/W	00h	49.3.34/ 2678
4002_8024	UART CEA709.1-B Beta1 Timer (UART1_B1T)	8	R/W	00h	49.3.35/ 2678
4002_8025	UART CEA709.1-B Secondary Delay Timer High (UART1_SDTH)	8	R/W	00h	49.3.36/ 2679
4002_8026	UART CEA709.1-B Secondary Delay Timer Low (UART1_SDTL)	8	R/W	00h	49.3.37/ 2679
4002_8027	UART CEA709.1-B Preamble (UART1_PRE)	8	R/W	00h	49.3.38/ 2679
4002_8028	UART CEA709.1-B Transmit Packet Length (UART1_TPL)	8	R/W	00h	49.3.39/ 2680
4002_8029	UART CEA709.1-B Interrupt Enable Register (UART1_IE)	8	R/W	00h	49.3.40/ 2680
4002_802A	UART CEA709.1-B WBASE (UART1_WB)	8	R/W	00h	49.3.41/ 2681
4002_802B	UART CEA709.1-B Status Register (UART1_S3)	8	R/W	00h	49.3.42/ 2682
4002_802C	UART CEA709.1-B Status Register (UART1_S4)	8	R/W	00h	49.3.43/ 2683
4002_802D	UART CEA709.1-B Received Packet Length (UART1_RPL)	8	R	00h	49.3.44/ 2684
4002_802E	UART CEA709.1-B Received Preamble Length (UART1_RPREL)	8	R	00h	49.3.45/ 2685
4002_802F	UART CEA709.1-B Collision Pulse Width (UART1_CPW)	8	R/W	00h	49.3.46/ 2685
4002_8030	UART CEA709.1-B Receive Indeterminate Time (UART1_RIDT)	8	R/W	00h	49.3.47/ 2685
4002_8031	UART CEA709.1-B Transmit Indeterminate Time (UART1_TIDT)	8	R/W	00h	49.3.48/ 2686

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_9000	UART Baud Rate Registers: High (UART2_BDH)	8	R/W	00h	49.3.1/2646
4002_9001	UART Baud Rate Registers: Low (UART2_BDL)	8	R/W	04h	49.3.2/2647
4002_9002	UART Control Register 1 (UART2_C1)	8	R/W	00h	49.3.3/2647
4002_9003	UART Control Register 2 (UART2_C2)	8	R/W	00h	49.3.4/2649
4002_9004	UART Status Register 1 (UART2_S1)	8	R	C0h	49.3.5/2651
4002_9005	UART Status Register 2 (UART2_S2)	8	R/W	00h	49.3.6/2654
4002_9006	UART Control Register 3 (UART2_C3)	8	R/W	00h	49.3.7/2656
4002_9007	UART Data Register (UART2_D)	8	R/W	00h	49.3.8/2657
4002_9008	UART Match Address Registers 1 (UART2_MA1)	8	R/W	00h	49.3.9/2658
4002_9009	UART Match Address Registers 2 (UART2_MA2)	8	R/W	00h	49.3.10/ 2659
4002_900A	UART Control Register 4 (UART2_C4)	8	R/W	00h	49.3.11/ 2659
4002_900B	UART Control Register 5 (UART2_C5)	8	R/W	00h	49.3.12/ 2660
4002_900C	UART Extended Data Register (UART2_ED)	8	R	00h	49.3.13/ 2661
4002_900D	UART Modem Register (UART2_MODEM)	8	R/W	00h	49.3.14/ 2662
4002_900E	UART Infrared Register (UART2_IR)	8	R/W	00h	49.3.15/ 2663
4002_9010	UART FIFO Parameters (UART2_PFIFO)	8	R/W	See section	49.3.16/ 2664
4002_9011	UART FIFO Control Register (UART2_CFIFO)	8	R/W	00h	49.3.17/ 2665
4002_9012	UART FIFO Status Register (UART2_SFIFO)	8	R/W	C0h	49.3.18/ 2666
4002_9013	UART FIFO Transmit Watermark (UART2_TWFIFO)	8	R/W	00h	49.3.19/ 2667
4002_9014	UART FIFO Transmit Count (UART2_TCFIFO)	8	R	00h	49.3.20/ 2668
4002_9015	UART FIFO Receive Watermark (UART2_RWFIFO)	8	R/W	01h	49.3.21/ 2668
4002_9016	UART FIFO Receive Count (UART2_RCFIFO)	8	R	00h	49.3.22/ 2669
4002_9018	UART 7816 Control Register (UART2_C7816)	8	R/W	00h	49.3.23/ 2669
4002_9019	UART 7816 Interrupt Enable Register (UART2_IE7816)	8	R/W	00h	49.3.24/ 2671
4002_901A	UART 7816 Interrupt Status Register (UART2_IS7816)	8	R/W	00h	49.3.25/ 2672
4002_901B	UART 7816 Wait Parameter Register (UART2_WP7816T0)	8	R/W	0Ah	49.3.26/ 2673

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_901B	UART 7816 Wait Parameter Register (UART2_WP7816T1)	8	R/W	0Ah	49.3.27/ 2674
4002_901C	UART 7816 Wait N Register (UART2_WN7816)	8	R/W	00h	49.3.28/ 2674
4002_901D	UART 7816 Wait FD Register (UART2_WF7816)	8	R/W	01h	49.3.29/ 2675
4002_901E	UART 7816 Error Threshold Register (UART2_ET7816)	8	R/W	00h	49.3.30/ 2675
4002_901F	UART 7816 Transmit Length Register (UART2_TL7816)	8	R/W	00h	49.3.31/ 2676
4002_9021	UART CEA709.1-B Control Register 6 (UART2_C6)	8	R/W	00h	49.3.32/ 2677
4002_9022	UART CEA709.1-B Packet Cycle Time Counter High (UART2_PCTH)	8	R/W	00h	49.3.33/ 2677
4002_9023	UART CEA709.1-B Packet Cycle Time Counter Low (UART2_PCTL)	8	R/W	00h	49.3.34/ 2678
4002_9024	UART CEA709.1-B Beta1 Timer (UART2_B1T)	8	R/W	00h	49.3.35/ 2678
4002_9025	UART CEA709.1-B Secondary Delay Timer High (UART2_SDTH)	8	R/W	00h	49.3.36/ 2679
4002_9026	UART CEA709.1-B Secondary Delay Timer Low (UART2_SDTL)	8	R/W	00h	49.3.37/ 2679
4002_9027	UART CEA709.1-B Preamble (UART2_PRE)	8	R/W	00h	49.3.38/ 2679
4002_9028	UART CEA709.1-B Transmit Packet Length (UART2_TPL)	8	R/W	00h	49.3.39/ 2680
4002_9029	UART CEA709.1-B Interrupt Enable Register (UART2_IE)	8	R/W	00h	49.3.40/ 2680
4002_902A	UART CEA709.1-B WBASE (UART2_WB)	8	R/W	00h	49.3.41/ 2681
4002_902B	UART CEA709.1-B Status Register (UART2_S3)	8	R/W	00h	49.3.42/ 2682
4002_902C	UART CEA709.1-B Status Register (UART2_S4)	8	R/W	00h	49.3.43/ 2683
4002_902D	UART CEA709.1-B Received Packet Length (UART2_RPL)	8	R	00h	49.3.44/ 2684
4002_902E	UART CEA709.1-B Received Preamble Length (UART2_RPREL)	8	R	00h	49.3.45/ 2685
4002_902F	UART CEA709.1-B Collision Pulse Width (UART2_CPW)	8	R/W	00h	49.3.46/ 2685
4002_9030	UART CEA709.1-B Receive Indeterminate Time (UART2_RIDT)	8	R/W	00h	49.3.47/ 2685
4002_9031	UART CEA709.1-B Transmit Indeterminate Time (UART2_TIDT)	8	R/W	00h	49.3.48/ 2686

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_A000	UART Baud Rate Registers: High (UART3_BDH)	8	R/W	00h	49.3.1/2646
4002_A001	UART Baud Rate Registers: Low (UART3_BDL)	8	R/W	04h	49.3.2/2647
4002_A002	UART Control Register 1 (UART3_C1)	8	R/W	00h	49.3.3/2647
4002_A003	UART Control Register 2 (UART3_C2)	8	R/W	00h	49.3.4/2649
4002_A004	UART Status Register 1 (UART3_S1)	8	R	C0h	49.3.5/2651
4002_A005	UART Status Register 2 (UART3_S2)	8	R/W	00h	49.3.6/2654
4002_A006	UART Control Register 3 (UART3_C3)	8	R/W	00h	49.3.7/2656
4002_A007	UART Data Register (UART3_D)	8	R/W	00h	49.3.8/2657
4002_A008	UART Match Address Registers 1 (UART3_MA1)	8	R/W	00h	49.3.9/2658
4002_A009	UART Match Address Registers 2 (UART3_MA2)	8	R/W	00h	49.3.10/ 2659
4002_A00A	UART Control Register 4 (UART3_C4)	8	R/W	00h	49.3.11/ 2659
4002_A00B	UART Control Register 5 (UART3_C5)	8	R/W	00h	49.3.12/ 2660
4002_A00C	UART Extended Data Register (UART3_ED)	8	R	00h	49.3.13/ 2661
4002_A00D	UART Modem Register (UART3_MODEM)	8	R/W	00h	49.3.14/ 2662
4002_A00E	UART Infrared Register (UART3_IR)	8	R/W	00h	49.3.15/ 2663
4002_A010	UART FIFO Parameters (UART3_PFIFO)	8	R/W	See section	49.3.16/ 2664
4002_A011	UART FIFO Control Register (UART3_CFIFO)	8	R/W	00h	49.3.17/ 2665
4002_A012	UART FIFO Status Register (UART3_SFIFO)	8	R/W	C0h	49.3.18/ 2666
4002_A013	UART FIFO Transmit Watermark (UART3_TWFIFO)	8	R/W	00h	49.3.19/ 2667
4002_A014	UART FIFO Transmit Count (UART3_TCFIFO)	8	R	00h	49.3.20/ 2668
4002_A015	UART FIFO Receive Watermark (UART3_RWFIFO)	8	R/W	01h	49.3.21/ 2668
4002_A016	UART FIFO Receive Count (UART3_RCFIFO)	8	R	00h	49.3.22/ 2669
4002_A018	UART 7816 Control Register (UART3_C7816)	8	R/W	00h	49.3.23/ 2669
4002_A019	UART 7816 Interrupt Enable Register (UART3_IE7816)	8	R/W	00h	49.3.24/ 2671
4002_A01A	UART 7816 Interrupt Status Register (UART3_IS7816)	8	R/W	00h	49.3.25/ 2672
4002_A01B	UART 7816 Wait Parameter Register (UART3_WP7816T0)	8	R/W	0Ah	49.3.26/ 2673

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_A01B	UART 7816 Wait Parameter Register (UART3_WP7816T1)	8	R/W	0Ah	49.3.27/ 2674
4002_A01C	UART 7816 Wait N Register (UART3_WN7816)	8	R/W	00h	49.3.28/ 2674
4002_A01D	UART 7816 Wait FD Register (UART3_WF7816)	8	R/W	01h	49.3.29/ 2675
4002_A01E	UART 7816 Error Threshold Register (UART3_ET7816)	8	R/W	00h	49.3.30/ 2675
4002_A01F	UART 7816 Transmit Length Register (UART3_TL7816)	8	R/W	00h	49.3.31/ 2676
4002_A021	UART CEA709.1-B Control Register 6 (UART3_C6)	8	R/W	00h	49.3.32/ 2677
4002_A022	UART CEA709.1-B Packet Cycle Time Counter High (UART3_PCTH)	8	R/W	00h	49.3.33/ 2677
4002_A023	UART CEA709.1-B Packet Cycle Time Counter Low (UART3_PCTL)	8	R/W	00h	49.3.34/ 2678
4002_A024	UART CEA709.1-B Beta1 Timer (UART3_B1T)	8	R/W	00h	49.3.35/ 2678
4002_A025	UART CEA709.1-B Secondary Delay Timer High (UART3_SDTH)	8	R/W	00h	49.3.36/ 2679
4002_A026	UART CEA709.1-B Secondary Delay Timer Low (UART3_SDTL)	8	R/W	00h	49.3.37/ 2679
4002_A027	UART CEA709.1-B Preamble (UART3_PRE)	8	R/W	00h	49.3.38/ 2679
4002_A028	UART CEA709.1-B Transmit Packet Length (UART3_TPL)	8	R/W	00h	49.3.39/ 2680
4002_A029	UART CEA709.1-B Interrupt Enable Register (UART3_IE)	8	R/W	00h	49.3.40/ 2680
4002_A02A	UART CEA709.1-B WBASE (UART3_WB)	8	R/W	00h	49.3.41/ 2681
4002_A02B	UART CEA709.1-B Status Register (UART3_S3)	8	R/W	00h	49.3.42/ 2682
4002_A02C	UART CEA709.1-B Status Register (UART3_S4)	8	R/W	00h	49.3.43/ 2683
4002_A02D	UART CEA709.1-B Received Packet Length (UART3_RPL)	8	R	00h	49.3.44/ 2684
4002_A02E	UART CEA709.1-B Received Preamble Length (UART3_RPREL)	8	R	00h	49.3.45/ 2685
4002_A02F	UART CEA709.1-B Collision Pulse Width (UART3_CPW)	8	R/W	00h	49.3.46/ 2685
4002_A030	UART CEA709.1-B Receive Indeterminate Time (UART3_RIDT)	8	R/W	00h	49.3.47/ 2685
4002_A031	UART CEA709.1-B Transmit Indeterminate Time (UART3_TIDT)	8	R/W	00h	49.3.48/ 2686

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_9000	UART Baud Rate Registers: High (UART4_BDH)	8	R/W	00h	49.3.1/2646
400A_9001	UART Baud Rate Registers: Low (UART4_BDL)	8	R/W	04h	49.3.2/2647
400A_9002	UART Control Register 1 (UART4_C1)	8	R/W	00h	49.3.3/2647
400A_9003	UART Control Register 2 (UART4_C2)	8	R/W	00h	49.3.4/2649
400A_9004	UART Status Register 1 (UART4_S1)	8	R	C0h	49.3.5/2651
400A_9005	UART Status Register 2 (UART4_S2)	8	R/W	00h	49.3.6/2654
400A_9006	UART Control Register 3 (UART4_C3)	8	R/W	00h	49.3.7/2656
400A_9007	UART Data Register (UART4_D)	8	R/W	00h	49.3.8/2657
400A_9008	UART Match Address Registers 1 (UART4_MA1)	8	R/W	00h	49.3.9/2658
400A_9009	UART Match Address Registers 2 (UART4_MA2)	8	R/W	00h	49.3.10/ 2659
400A_900A	UART Control Register 4 (UART4_C4)	8	R/W	00h	49.3.11/ 2659
400A_900B	UART Control Register 5 (UART4_C5)	8	R/W	00h	49.3.12/ 2660
400A_900C	UART Extended Data Register (UART4_ED)	8	R	00h	49.3.13/ 2661
400A_900D	UART Modem Register (UART4_MODEM)	8	R/W	00h	49.3.14/ 2662
400A_900E	UART Infrared Register (UART4_IR)	8	R/W	00h	49.3.15/ 2663
400A_9010	UART FIFO Parameters (UART4_PFIFO)	8	R/W	See section	49.3.16/ 2664
400A_9011	UART FIFO Control Register (UART4_CFIFO)	8	R/W	00h	49.3.17/ 2665
400A_9012	UART FIFO Status Register (UART4_SFIFO)	8	R/W	C0h	49.3.18/ 2666
400A_9013	UART FIFO Transmit Watermark (UART4_TWFIFO)	8	R/W	00h	49.3.19/ 2667
400A_9014	UART FIFO Transmit Count (UART4_TCFIFO)	8	R	00h	49.3.20/ 2668
400A_9015	UART FIFO Receive Watermark (UART4_RWFIFO)	8	R/W	01h	49.3.21/ 2668
400A_9016	UART FIFO Receive Count (UART4_RCFIFO)	8	R	00h	49.3.22/ 2669
400A_9018	UART 7816 Control Register (UART4_C7816)	8	R/W	00h	49.3.23/ 2669
400A_9019	UART 7816 Interrupt Enable Register (UART4_IE7816)	8	R/W	00h	49.3.24/ 2671
400A_901A	UART 7816 Interrupt Status Register (UART4_IS7816)	8	R/W	00h	49.3.25/ 2672
400A_901B	UART 7816 Wait Parameter Register (UART4_WP7816T0)	8	R/W	0Ah	49.3.26/ 2673

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_901B	UART 7816 Wait Parameter Register (UART4_WP7816T1)	8	R/W	0Ah	49.3.27/ 2674
400A_901C	UART 7816 Wait N Register (UART4_WN7816)	8	R/W	00h	49.3.28/ 2674
400A_901D	UART 7816 Wait FD Register (UART4_WF7816)	8	R/W	01h	49.3.29/ 2675
400A_901E	UART 7816 Error Threshold Register (UART4_ET7816)	8	R/W	00h	49.3.30/ 2675
400A_901F	UART 7816 Transmit Length Register (UART4_TL7816)	8	R/W	00h	49.3.31/ 2676
400A_9021	UART CEA709.1-B Control Register 6 (UART4_C6)	8	R/W	00h	49.3.32/ 2677
400A_9022	UART CEA709.1-B Packet Cycle Time Counter High (UART4_PCTH)	8	R/W	00h	49.3.33/ 2677
400A_9023	UART CEA709.1-B Packet Cycle Time Counter Low (UART4_PCTL)	8	R/W	00h	49.3.34/ 2678
400A_9024	UART CEA709.1-B Beta1 Timer (UART4_B1T)	8	R/W	00h	49.3.35/ 2678
400A_9025	UART CEA709.1-B Secondary Delay Timer High (UART4_SDTH)	8	R/W	00h	49.3.36/ 2679
400A_9026	UART CEA709.1-B Secondary Delay Timer Low (UART4_SDTL)	8	R/W	00h	49.3.37/ 2679
400A_9027	UART CEA709.1-B Preamble (UART4_PRE)	8	R/W	00h	49.3.38/ 2679
400A_9028	UART CEA709.1-B Transmit Packet Length (UART4_TPL)	8	R/W	00h	49.3.39/ 2680
400A_9029	UART CEA709.1-B Interrupt Enable Register (UART4_IE)	8	R/W	00h	49.3.40/ 2680
400A_902A	UART CEA709.1-B WBASE (UART4_WB)	8	R/W	00h	49.3.41/ 2681
400A_902B	UART CEA709.1-B Status Register (UART4_S3)	8	R/W	00h	49.3.42/ 2682
400A_902C	UART CEA709.1-B Status Register (UART4_S4)	8	R/W	00h	49.3.43/ 2683
400A_902D	UART CEA709.1-B Received Packet Length (UART4_RPL)	8	R	00h	49.3.44/ 2684
400A_902E	UART CEA709.1-B Received Preamble Length (UART4_RPREL)	8	R	00h	49.3.45/ 2685
400A_902F	UART CEA709.1-B Collision Pulse Width (UART4_CPW)	8	R/W	00h	49.3.46/ 2685
400A_9030	UART CEA709.1-B Receive Indeterminate Time (UART4_RIDT)	8	R/W	00h	49.3.47/ 2685
400A_9031	UART CEA709.1-B Transmit Indeterminate Time (UART4_TIDT)	8	R/W	00h	49.3.48/ 2686

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_A000	UART Baud Rate Registers: High (UART5_BDH)	8	R/W	00h	49.3.1/2646
400A_A001	UART Baud Rate Registers: Low (UART5_BDL)	8	R/W	04h	49.3.2/2647
400A_A002	UART Control Register 1 (UART5_C1)	8	R/W	00h	49.3.3/2647
400A_A003	UART Control Register 2 (UART5_C2)	8	R/W	00h	49.3.4/2649
400A_A004	UART Status Register 1 (UART5_S1)	8	R	C0h	49.3.5/2651
400A_A005	UART Status Register 2 (UART5_S2)	8	R/W	00h	49.3.6/2654
400A_A006	UART Control Register 3 (UART5_C3)	8	R/W	00h	49.3.7/2656
400A_A007	UART Data Register (UART5_D)	8	R/W	00h	49.3.8/2657
400A_A008	UART Match Address Registers 1 (UART5_MA1)	8	R/W	00h	49.3.9/2658
400A_A009	UART Match Address Registers 2 (UART5_MA2)	8	R/W	00h	49.3.10/ 2659
400A_A00A	UART Control Register 4 (UART5_C4)	8	R/W	00h	49.3.11/ 2659
400A_A00B	UART Control Register 5 (UART5_C5)	8	R/W	00h	49.3.12/ 2660
400A_A00C	UART Extended Data Register (UART5_ED)	8	R	00h	49.3.13/ 2661
400A_A00D	UART Modem Register (UART5_MODEM)	8	R/W	00h	49.3.14/ 2662
400A_A00E	UART Infrared Register (UART5_IR)	8	R/W	00h	49.3.15/ 2663
400A_A010	UART FIFO Parameters (UART5_PFIFO)	8	R/W	See section	49.3.16/ 2664
400A_A011	UART FIFO Control Register (UART5_CFIFO)	8	R/W	00h	49.3.17/ 2665
400A_A012	UART FIFO Status Register (UART5_SFIFO)	8	R/W	C0h	49.3.18/ 2666
400A_A013	UART FIFO Transmit Watermark (UART5_TWFIFO)	8	R/W	00h	49.3.19/ 2667
400A_A014	UART FIFO Transmit Count (UART5_TCFIFO)	8	R	00h	49.3.20/ 2668
400A_A015	UART FIFO Receive Watermark (UART5_RWFIFO)	8	R/W	01h	49.3.21/ 2668
400A_A016	UART FIFO Receive Count (UART5_RCFIFO)	8	R	00h	49.3.22/ 2669
400A_A018	UART 7816 Control Register (UART5_C7816)	8	R/W	00h	49.3.23/ 2669
400A_A019	UART 7816 Interrupt Enable Register (UART5_IE7816)	8	R/W	00h	49.3.24/ 2671
400A_A01A	UART 7816 Interrupt Status Register (UART5_IS7816)	8	R/W	00h	49.3.25/ 2672
400A_A01B	UART 7816 Wait Parameter Register (UART5_WP7816T0)	8	R/W	0Ah	49.3.26/ 2673

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400A_A01B	UART 7816 Wait Parameter Register (UART5_WP7816T1)	8	R/W	0Ah	49.3.27/ 2674
400A_A01C	UART 7816 Wait N Register (UART5_WN7816)	8	R/W	00h	49.3.28/ 2674
400A_A01D	UART 7816 Wait FD Register (UART5_WF7816)	8	R/W	01h	49.3.29/ 2675
400A_A01E	UART 7816 Error Threshold Register (UART5_ET7816)	8	R/W	00h	49.3.30/ 2675
400A_A01F	UART 7816 Transmit Length Register (UART5_TL7816)	8	R/W	00h	49.3.31/ 2676
400A_A021	UART CEA709.1-B Control Register 6 (UART5_C6)	8	R/W	00h	49.3.32/ 2677
400A_A022	UART CEA709.1-B Packet Cycle Time Counter High (UART5_PCTH)	8	R/W	00h	49.3.33/ 2677
400A_A023	UART CEA709.1-B Packet Cycle Time Counter Low (UART5_PCTL)	8	R/W	00h	49.3.34/ 2678
400A_A024	UART CEA709.1-B Beta1 Timer (UART5_B1T)	8	R/W	00h	49.3.35/ 2678
400A_A025	UART CEA709.1-B Secondary Delay Timer High (UART5_SDTH)	8	R/W	00h	49.3.36/ 2679
400A_A026	UART CEA709.1-B Secondary Delay Timer Low (UART5_SDTL)	8	R/W	00h	49.3.37/ 2679
400A_A027	UART CEA709.1-B Preamble (UART5_PRE)	8	R/W	00h	49.3.38/ 2679
400A_A028	UART CEA709.1-B Transmit Packet Length (UART5_TPL)	8	R/W	00h	49.3.39/ 2680
400A_A029	UART CEA709.1-B Interrupt Enable Register (UART5_IE)	8	R/W	00h	49.3.40/ 2680
400A_A02A	UART CEA709.1-B WBASE (UART5_WB)	8	R/W	00h	49.3.41/ 2681
400A_A02B	UART CEA709.1-B Status Register (UART5_S3)	8	R/W	00h	49.3.42/ 2682
400A_A02C	UART CEA709.1-B Status Register (UART5_S4)	8	R/W	00h	49.3.43/ 2683
400A_A02D	UART CEA709.1-B Received Packet Length (UART5_RPL)	8	R	00h	49.3.44/ 2684
400A_A02E	UART CEA709.1-B Received Preamble Length (UART5_RPREL)	8	R	00h	49.3.45/ 2685
400A_A02F	UART CEA709.1-B Collision Pulse Width (UART5_CPW)	8	R/W	00h	49.3.46/ 2685
400A_A030	UART CEA709.1-B Receive Indeterminate Time (UART5_RIDT)	8	R/W	00h	49.3.47/ 2685
400A_A031	UART CEA709.1-B Transmit Indeterminate Time (UART5_TIDT)	8	R/W	00h	49.3.48/ 2686

49.3.1 UART Baud Rate Registers: High (UARTx_BDH)

This register, along with the BDL register, controls the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting (SBR[12:0]), first write to BDH to buffer the high half of the new value and then write to BDL. The working value in BDH does not change until BDL is written.

BDL is reset to a nonzero value, but after reset, the baud rate generator remains disabled until the first time the receiver or transmitter is enabled, that is, when C2[RE] or C2[TE] is set.

Address: Base address + 0h offset

Bit	7	6	5	4	3	2	1	0
Read	LBKDIE	RXEDGIE	0	SBR				
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_BDH field descriptions

Field	Description
7 LBKDIE	LIN Break Detect Interrupt Enable Enables the LIN break detect flag, LBKDIF, to generate interrupt requests based on the state of LBKDDMAS. 0 LBKDIF interrupt requests disabled. 1 LBKDIF interrupt requests enabled.
6 RXEDGIE	RxD Input Active Edge Interrupt Enable Enables the receive input active edge, RXEDGIF, to generate interrupt requests. 0 Hardware interrupts from RXEDGIF disabled using polling. 1 RXEDGIF interrupt request enabled.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 SBR	UART Baud Rate Bits The baud rate for the UART is determined by the 13 SBR fields. See Baud rate generation for details. **NOTE:** • The baud rate generator is disabled until C2[TE] or C2[RE] is set for the first time after reset.The baud rate generator is disabled when SBR = 0. • Writing to BDH has no effect without writing to BDL, because writing to BDH puts the data in a temporary location until BDL is written.

49.3.2 UART Baud Rate Registers: Low (UARTx_BDL)

This register, along with the BDH register, controls the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting, SBR[12:0], first write to BDH to buffer the high half of the new value and then write to BDL. The working value in BDH does not change until BDL is written. BDL is reset to a nonzero value, but after reset, the baud rate generator remains disabled until the first time the receiver or transmitter is enabled, that is, when C2[RE] or C2[TE] is set.

Address: Base address + 1h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	SBR							
Reset	0	0	0	0	0	1	0	0

UARTx_BDL field descriptions

Field	Description
7–0 SBR	UART Baud Rate Bits The baud rate for the UART is determined by the 13 SBR fields. See Baud rate generation for details. **NOTE:** • The baud rate generator is disabled until C2[TE] or C2[RE] is set for the first time after reset.The baud rate generator is disabled when SBR = 0. • Writing to BDH has no effect without writing to BDL, because writing to BDH puts the data in a temporary location until BDL is written. • When the 1/32 narrow pulse width is selected for infrared (IrDA), the baud rate fields must be even, the least significant bit is 0. See MODEM register for more details.

49.3.3 UART Control Register 1 (UARTx_C1)

This read/write register controls various optional features of the UART system.

Address: Base address + 2h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	LOOPS	Reserved	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

UARTx_C1 field descriptions

Field	Description
7 LOOPS	Loop Mode Select When LOOPS is set, the RxD pin is disconnected from the UART and the transmitter output is internally connected to the receiver input. The transmitter and the receiver must be enabled to use the loop function.

Table continues on the next page...

UARTx_C1 field descriptions (continued)

Field	Description
	0 Normal operation. 1 Loop mode where transmitter output is internally connected to receiver input. The receiver input is determined by RSRC.
6 Reserved	Reserved. This field is reserved.
5 RSRC	Receiver Source Select This field has no meaning or effect unless the LOOPS field is set. When LOOPS is set, the RSRC field determines the source for the receiver shift register input. 0 Selects internal loop back mode. The receiver input is internally connected to transmitter output. 1 Single wire UART mode where the receiver input is connected to the transmit pin input signal.
4 M	9-bit or 8-bit Mode Select This field must be set when C7816[ISO_7816E] is set/enabled. 0 Normal—start + 8 data bits (MSB/LSB first as determined by MSBF) + stop. 1 Use—start + 9 data bits (MSB/LSB first as determined by MSBF) + stop.
3 WAKE	Receiver Wakeup Method Select Determines which condition wakes the UART: • Address mark in the most significant bit position of a received data character, or • An idle condition on the receive pin input signal. 0 Idle line wakeup. 1 Address mark wakeup.
2 ILT	Idle Line Type Select Determines when the receiver starts counting logic 1s as idle character bits. The count begins either after a valid start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit can cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. **NOTE:** • In case the UART is programmed with ILT = 1, a logic of 1'b0 is automatically shifted after a received stop bit, therefore resetting the idle count. • In case the UART is programmed for IDLE line wakeup (RWU = 1 and WAKE = 0), ILT has no effect on when the receiver starts counting logic 1s as idle character bits. In idle line wakeup, an idle character is recognized at anytime the receiver sees 10, 11, or 12 1s depending on the M, PE, and C4[M10] fields. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	Parity Enable Enables the parity function. When parity is enabled, parity function inserts a parity bit in the bit position immediately preceding the stop bit. This field must be set when C7816[ISO_7816E] is set/enabled. 0 Parity function disabled. 1 Parity function enabled.
0 PT	Parity Type

Table continues on the next page...

UARTx_C1 field descriptions (continued)

Field	Description
	Determines whether the UART generates and checks for even parity or odd parity. With even parity, an even number of 1s clears the parity bit and an odd number of 1s sets the parity bit. With odd parity, an odd number of 1s clears the parity bit and an even number of 1s sets the parity bit. This field must be cleared when C7816[ISO_7816E] is set/enabled. 0 Even parity. 1 Odd parity.

49.3.4 UART Control Register 2 (UARTx_C2)

This register can be read or written at any time.

Address: Base address + 3h offset

Bit	7	6	5	4	3	2	1	0
Read Write	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

UARTx_C2 field descriptions

Field	Description
7 TIE	Transmitter Interrupt or DMA Transfer Enable. Enables S1[TDRE] to generate interrupt requests or DMA transfer requests, based on the state of C5[TDMAS]. **NOTE:** If C2[TIE] and C5[TDMAS] are both set, then TCIE must be cleared, and D[D] must not be written unless servicing a DMA request. 0 TDRE interrupt and DMA transfer requests disabled. 1 TDRE interrupt or DMA transfer requests enabled.
6 TCIE	Transmission Complete Interrupt Enable Enables the transmission complete flag, S1[TC], to generate interrupt requests . 0 TC interrupt requests disabled. 1 TC interrupt requests enabled.
5 RIE	Receiver Full Interrupt or DMA Transfer Enable Enables S1[RDRF] to generate interrupt requests or DMA transfer requests, based on the state of C5[RDMAS]. 0 RDRF interrupt and DMA transfer requests disabled. 1 RDRF interrupt or DMA transfer requests enabled.
4 ILIE	Idle Line Interrupt Enable Enables the idle line flag, S1[IDLE], to generate interrupt requests

Table continues on the next page...

UARTx_C2 field descriptions (continued)

Field	Description
	0 IDLE interrupt requests disabled. 1 IDLE interrupt requests enabled.
3 TE	Transmitter Enable Enables the UART transmitter. TE can be used to queue an idle preamble by clearing and then setting TE. When C7816[ISO_7816E] is set/enabled and C7816[TTYPE] = 1, this field is automatically cleared after the requested block has been transmitted. This condition is detected when TL7816[TLEN] = 0 and four additional characters are transmitted. 0 Transmitter off. 1 Transmitter on.
2 RE	Receiver Enable Enables the UART receiver. 0 Receiver off. 1 Receiver on.
1 RWU	Receiver Wakeup Control This field can be set to place the UART receiver in a standby state. RWU automatically clears when an RWU event occurs, that is, an IDLE event when C1[WAKE] is clear or an address match when C1[WAKE] is set. This field must be cleared when C7816[ISO_7816E] is set. **NOTE:** RWU must be set only with C1[WAKE] = 0 (wakeup on idle) if the channel is currently not idle. This can be determined by S2[RAF]. If the flag is set to wake up an IDLE event and the channel is already idle, it is possible that the UART will discard data. This is because the data must be received or a LIN break detected after an IDLE is detected before IDLE is allowed to reasserted. 0 Normal operation. 1 RWU enables the wakeup function and inhibits further receiver interrupt requests. Normally, hardware wakes the receiver by automatically clearing RWU.
0 SBK	Send Break Toggling SBK sends one break character from the following: See Transmitting break characters for the number of logic 0s for the different configurations. Toggling implies clearing the SBK field before the break character has finished transmitting. As long as SBK is set, the transmitter continues to send complete break characters (10, 11, or 12 bits, or 13 or 14 bits). Ensure that C2[TE] is asserted atleast 1 clock before assertion of this bit. • 10, 11, or 12 logic 0s if S2[BRK13] is cleared • 13 or 14 logic 0s if S2[BRK13] is set. This field must be cleared when C7816[ISO_7816E] is set. 0 Normal transmitter operation. 1 Queue break characters to be sent.

49.3.5 UART Status Register 1 (UARTx_S1)

The S1 register provides inputs to the MCU for generation of UART interrupts or DMA requests. This register can also be polled by the MCU to check the status of its fields. To clear a flag, the status register should be read followed by a read or write to D register, depending on the interrupt flag type. Other instructions can be executed between the two steps as long the handling of I/O is not compromised, but the order of operations is important for flag clearing. When a flag is configured to trigger a DMA request, assertion of the associated DMA done signal from the DMA controller clears the flag.

NOTE

- If the condition that results in the assertion of the flag, interrupt, or DMA request is not resolved prior to clearing the flag, the flag, and interrupt/DMA request, reasserts. For example, if the DMA or interrupt service routine fails to write sufficient data to the transmit buffer to raise it above the watermark level, the flag reasserts and generates another interrupt or DMA request.
- Reading an empty data register to clear one of the flags of the S1 register causes the FIFO pointers to become misaligned. A receive FIFO flush reinitializes the pointers. A better way to prevent this situation is to always leave one byte in FIFO and this byte will be read eventually in clearing the flag bit.

Address: Base address + 4h offset

Bit	7	6	5	4	3	2	1	0
Read	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write								
Reset	1	1	0	0	0	0	0	0

UARTx_S1 field descriptions

Field	Description
7 TDRE	Transmit Data Register Empty Flag TDRE will set when the number of datawords in the transmit buffer (D and C3[T8])is equal to or less than the number indicated by TWFIFO[TXWATER]. A character that is in the process of being transmitted is not included in the count. To clear TDRE, read S1 when TDRE is set and then write to the UART data register (D). For more efficient interrupt servicing, all data except the final value to be written to the buffer must be written to D/C3[T8]. Then S1 can be read before writing the final data value, resulting in the clearing of the TRDE flag. This is more efficient because the TDRE reasserts until the watermark has been exceeded. So, attempting to clear the TDRE with every write will be ineffective until sufficient data has been written.

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	0 The amount of data in the transmit buffer is greater than the value indicated by TWFIFO[TXWATER]. 1 The amount of data in the transmit buffer is less than or equal to the value indicated by TWFIFO[TXWATER] at some point in time since the flag has been cleared.
6 TC	Transmit Complete Flag TC is set when the transmit buffer is empty and no data, preamble, or break character is being transmitted. When TC is set, the transmit data output signal becomes idle (logic 1). TC is cleared by reading S1 with TC set and then doing one of the following: When C7816[ISO_7816E] is set/enabled, this field is set after any NACK signal has been received, but prior to any corresponding guard times expiring.When C6[EN709] is set/enabled, this flag is not set on transmit packet completion. • Writing to D to transmit new data. • Queuing a preamble by clearing and then setting C2[TE]. • Queuing a break character by writing 1 to SBK in C2. 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete).
5 RDRF	Receive Data Register Full Flag RDRF is set when the number of datawords in the receive buffer is equal to or more than the number indicated by RWFIFO[RXWATER]. A dataword that is in the process of being received is not included in the count. To clear RDRF, read S1 when RDRF is set and then read D. For more efficient interrupt and DMA operation, read all data except the final value from the buffer, using D/C3[T8]/ED. Then read S1 and the final data value, resulting in the clearing of the RDRF flag. Even if RDRF is set, data will continue to be received until an overrun condition occurs.RDRF is prevented from setting while S2[LBKDE] is set. Additionally, when S2[LBKDE] is set, the received datawords are stored in the receive buffer but over-write each other. 0 The number of datawords in the receive buffer is less than the number indicated by RXWATER. 1 The number of datawords in the receive buffer is equal to or greater than the number indicated by RXWATER at some point in time since this flag was last cleared.
4 IDLE	Idle Line Flag After the IDLE flag is cleared, a frame must be received (although not necessarily stored in the data buffer, for example if C2[RWU] is set), or a LIN break character must set the S2[LBKDIF] flag before an idle condition can set the IDLE flag. To clear IDLE, read UART status S1 with IDLE set and then read D. IDLE is set when either of the following appear on the receiver input: • 10 consecutive logic 1s if C1[M] = 0 • 11 consecutive logic 1s if C1[M] = 1 and C4[M10] = 0 • 12 consecutive logic 1s if C1[M] = 1, C4[M10] = 1, and C1[PE] = 1 Idle detection is not supported when7816Eor EN709is set/enabled and hence this flag is ignored. **NOTE:** When RWU is set and WAKE is cleared, an idle line condition sets the IDLE flag if RWUID is set, else the IDLE flag does not become set. 0 Receiver input is either active now or has never become active since the IDLE flag was last cleared. 1 Receiver input has become idle or the flag has not been cleared since it last asserted.
3 OR	Receiver Overrun Flag OR is set when software fails to prevent the receive data register from overflowing with data. The OR bit is set immediately after the stop bit has been completely received for the dataword that overflows the buffer and all the other error flags (FE, NF, and PF) are prevented from setting. The data in the shift register is lost, but the data already in the UART data registers is not affected. If the OR flag is set, no data is stored in the data buffer even if sufficient room exists. Additionally, while the OR flag is set, the RDRF and IDLE flags are blocked from asserting, that is, transition from an inactive to an active state. To clear OR, read

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	S1 when OR is set and then read D. See functional description for more details regarding the operation of the OR bit.If LBKDE is enabled and a LIN Break is detected, the OR field asserts if S2[LBKDIF] is not cleared before the next data character is received. In 7816 mode, it is possible to configure a NACK to be returned by programing C7816[ONACK]. 0 No overrun has occurred since the last time the flag was cleared. 1 Overrun has occurred or the overrun flag has not been cleared since the last overrun occured.
2 NF	Noise Flag NF is set when the UART detects noise on the receiver input. NF does not become set in the case of an overrun or while the LIN break detect feature is enabled (S2[LBKDE] = 1). When NF is set, it indicates only that a dataword has been received with noise since the last time it was cleared. There is no guarantee that the first dataword read from the receive buffer has noise or that there is only one dataword in the buffer that was received with noise unless the receive buffer has a depth of one. To clear NF, read S1 and then read D. When EN709 is set/enabled, noise flag is not set. 0 No noise detected since the last time this flag was cleared. If the receive buffer has a depth greater than 1 then there may be data in the receiver buffer that was received with noise. 1 At least one dataword was received with noise detected since the last time the flag was cleared.
1 FE	Framing Error Flag FE is set when a logic 0 is accepted as the stop bit. FE does not set in the case of an overrun or while the LIN break detect feature is enabled (S2[LBKDE] = 1). FE inhibits further data reception until it is cleared. To clear FE, read S1 with FE set and then read D. The last data in the receive buffer represents the data that was received with the frame error enabled. Framing errors are not supported when 7816E is set/ enabled. However, if this flag is set, data is still not received in 7816 mode.Framing errors are not supported in 709 mode. 0 No framing error detected. 1 Framing error.
0 PF	Parity Error Flag PF is set when PE is set and the parity of the received data does not match its parity bit. The PF is not set in the case of an overrun condition. When PF is set, it indicates only that a dataword was received with parity error since the last time it was cleared. There is no guarantee that the first dataword read from the receive buffer has a parity error or that there is only one dataword in the buffer that was received with a parity error, unless the receive buffer has a depth of one. To clear PF, read S1 and then read D., S2[LBKDE] is disabled,Within the receive buffer structure the received dataword is tagged if it is received with a parity error. This information is available by reading the ED register prior to reading the D register. 0 No parity error detected since the last time this flag was cleared. If the receive buffer has a depth greater than 1, then there may be data in the receive buffer what was received with a parity error. 1 At least one dataword was received with a parity error since the last time this flag was cleared.

49.3.6 UART Status Register 2 (UARTx_S2)

The S2 register provides inputs to the MCU for generation of UART interrupts or DMA requests. Also, this register can be polled by the MCU to check the status of these bits. This register can be read or written at any time, with the exception of the MSBF and RXINV bits, which should be changed by the user only between transmit and receive packets.

Address: Base address + 5h offset

Bit	7	6	5	4	3	2	1	0
Read	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF
Write	w1c	w1c						
Reset	0	0	0	0	0	0	0	0

UARTx_S2 field descriptions

Field	Description
7 LBKDIF	LIN Break Detect Interrupt Flag LBKDIF is set when LBKDE is set and a LIN break character is detected on the receiver input. The LIN break characters are 11 consecutive logic 0s if C1[M] = 0 or 12 consecutive logic 0s if C1[M] = 1. LBKDIF is set after receiving the last LIN break character. LBKDIF is cleared by writing a 1 to it. 0 No LIN break character detected. 1 LIN break character detected.
6 RXEDGIF	RxD Pin Active Edge Interrupt Flag RXEDGIF is set when an active edge occurs on the RxD pin. The active edge is falling if RXINV = 0, and rising if RXINV=1. RXEDGIF is cleared by writing a 1 to it. See for additional details. RXEDGIF description **NOTE:** The active edge is detected only in two wire mode and on receiving data coming from the RxD pin. 0 No active edge on the receive pin has occurred. 1 An active edge on the receive pin has occurred.
5 MSBF	Most Significant Bit First Setting this field reverses the order of the bits that are transmitted and received on the wire. This field does not affect the polarity of the bits, the location of the parity bit, or the location of the start or stop bits. This field is automatically set when C7816[INIT] and C7816[ISO7816E] are enabled and an initial character is detected in T = 0 protocol mode.In EN709 mode, this field affects the order of bits the same way as it does in normal mode. 0 LSB (bit0) is the first bit that is transmitted following the start bit. Further, the first bit received after the start bit is identified as bit0. 1 MSB (bit8, bit7 or bit6) is the first bit that is transmitted following the start bit, depending on the setting of C1[M] and C1[PE]. Further, the first bit received after the start bit is identified as bit8, bit7, or bit6, depending on the setting of C1[M] and C1[PE].
4 RXINV	Receive Data Inversion

Table continues on the next page...

UARTx_S2 field descriptions (continued)

Field	Description
	Setting this field reverses the polarity of the received data input. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity. A zero is represented by a short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity.This field is automatically set when C7816[INIT] and C7816[ISO7816E] are enabled and an initial character is detected in T = 0 protocol mode.In EN709 mode, this bit affects the polarity of bits the same as it does in normal mode. **NOTE:** Setting RXINV inverts the RxD input for data bits, start and stop bits, break, and idle. When C7816[ISO7816E] is set/enabled, only the data bits and the parity bit are inverted. 0 Receive data is not inverted. 1 Receive data is inverted.
3 RWUID	Receive Wakeup Idle Detect When RWU is set and WAKE is cleared, this field controls whether the idle character that wakes the receiver sets S1[IDLE]. This field must be cleared when C7816[ISO7816E] is set/enabled. 0 S1[IDLE] is not set upon detection of an idle character. 1 S1[IDLE] is set upon detection of an idle character.
2 BRK13	Break Transmit Character Length Determines whether the transmit break character is 10, 11, or 12 bits long, or 13 or 14 bits long. See for the length of the break character for the different configurations. The detection of a framing error is not affected by this field. Transmitting break characters 0 Break character is 10, 11, or 12 bits long. 1 Break character is 13 or 14 bits long.
1 LBKDE	LIN Break Detection Enable Enables the LIN Break detection feature. While LBKDE is set, S1[RDRF], S1[NF], S1[FE], and S1[PF] are prevented from setting. When LBKDE is set, see . Overrun operationLBKDE must be cleared when C7816[ISO7816E] is set. 0 Break character detection is disabled. 1 Break character is detected at length of 11 bit times if C1[M] = 0 or 12 bits time if C1[M] = 1.
0 RAF	Receiver Active Flag RAF is set when the UART receiver detects a logic 0 during the RT1 time period of the start bit search. RAF is cleared when the receiver detects an idle character when C7816[ISO7816E] is cleared/disabled. When C7816[ISO7816E] is enabled, the RAF is cleared if the C7816[TTYPE] = 0 expires or the C7816[TTYPE] = 1 expires. **NOTE:** In case C7816[ISO7816E] is set and C7816[TTYPE] = 0, it is possible to configure the guard time to 12. However, if a NACK is required to be transmitted, the data transfer actually takes 13 ETU with the 13th ETU slot being a inactive buffer. Therefore, in this situation, the RAF may deassert one ETU prior to actually being inactive. 0 UART receiver idle/inactive waiting for a start bit. 1 UART receiver active, RxD input not idle.

49.3.7 UART Control Register 3 (UARTx_C3)

Writing R8 does not have any effect. TXDIR and TXINV can be changed only between transmit and receive packets.

Address: Base address + 6h offset

Bit	7	6	5	4	3	2	1	0
Read	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_C3 field descriptions

Field	Description
7 R8	Received Bit 8 R8 is the ninth data bit received when the UART is configured for 9-bit data format, that is, if C1[M] = 1 or C4[M10] = 1. The R8 value corresponds to the current data value in the UARTx_D register. To read the 9th bit, read the value of UARTx_C3[R8], then read the UARTx_D register.
6 T8	Transmit Bit 8 T8 is the ninth data bit transmitted when the UART is configured for 9-bit data format, that is, if C1[M] = 1 or C4[M10] = 1. **NOTE:** If the value of T8 is the same as in the previous transmission, T8 does not have to be rewritten. The same value is transmitted until T8 is rewritten. To correctly transmit the 9th bit, write UARTx_C3[T8] to the desired value, then write the UARTx_D register with the remaining data.
5 TXDIR	Transmitter Pin Data Direction in Single-Wire mode Determines whether the TXD pin is used as an input or output in the single-wire mode of operation. This field is relevant only to the single wire mode. When C7816[ISO7816E] is set/enabled and C7816[TTYPE] = 1, this field is automatically cleared after the requested block is transmitted. This condition is detected when TL7816[TLEN] = 0 and 4 additional characters are transmitted. Additionally, if C7816[ISO7816E] is set/enabled and C7816[TTYPE] = 0 and a NACK is being transmitted, the hardware automatically overrides this field as needed. In this situation, TXDIR does not reflect the temporary state associated with the NACK. 0 TXD pin is an input in single wire mode. 1 TXD pin is an output in single wire mode.
4 TXINV	Transmit Data Inversion. Setting this field reverses the polarity of the transmitted data output. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity, and a zero is represented by short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity.This field is automatically set when C7816[INIT] and C7816[ISO7816E] are enabled and an initial character is detected in T = 0 protocol mode. **NOTE:** Setting TXINV inverts all transmitted values, including idle, break, start, and stop bits. In loop mode, if TXINV is set, the receiver gets the transmit inversion bit when RXINV is disabled. When C7816[ISO7816E] is set/enabled then only the transmitted data bits and parity bit are inverted.

Table continues on the next page...

UARTx_C3 field descriptions (continued)

Field	Description
	0 Transmit data is not inverted. 1 Transmit data is inverted.
3 ORIE	Overrun Error Interrupt Enable Enables the overrun error flag, S1[OR], to generate interrupt requests. 0 OR interrupts are disabled. 1 OR interrupt requests are enabled.
2 NEIE	Noise Error Interrupt Enable Enables the noise flag, S1[NF], to generate interrupt requests. 0 NF interrupt requests are disabled. 1 NF interrupt requests are enabled.
1 FEIE	Framing Error Interrupt Enable Enables the framing error flag, S1[FE], to generate interrupt requests. 0 FE interrupt requests are disabled. 1 FE interrupt requests are enabled.
0 PEIE	Parity Error Interrupt Enable Enables the parity error flag, S1[PF], to generate interrupt requests. 0 PF interrupt requests are disabled. 1 PF interrupt requests are enabled.

49.3.8 UART Data Register (UARTx_D)

This register is actually two separate registers. Reads return the contents of the read-only receive data register and writes go to the write-only transmit data register.

NOTE

- In 8-bit or 9-bit data format, only UART data register (D) needs to be accessed to clear the S1[RDRF] bit (assuming receiver buffer level is less than RWFIFO[RXWATER]). The C3 register needs to be read, prior to the D register, only if the ninth bit of data needs to be captured. Similarly, the ED register needs to be read, prior to the D register, only if the additional flag data for the dataword needs to be captured.
- In the normal 8-bit mode (M bit cleared) if the parity is enabled, you get seven data bits and one parity bit. That one parity bit is loaded into the D register. So, for the data

bits, mask off the parity bit from the value you read out of this register.

- When transmitting in 9-bit data format and using 8-bit write instructions, write first to transmit bit 8 in UART control register 3 (C3[T8]), then D. A write to C3[T8] stores the data in a temporary register. If D register is written first, and then the new data on data bus is stored in D, the temporary value written by the last write to C3[T8] gets stored in the C3[T8] register.

Address: Base address + 7h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	RT							
Reset	0	0	0	0	0	0	0	0

UARTx_D field descriptions

Field	Description
7–0 RT	Reads return the contents of the read-only receive data register and writes go to the write-only transmit data register.

49.3.9 UART Match Address Registers 1 (UARTx_MA1)

The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated C4[MAEN] field is set. If a match occurs, the following data is transferred to the data register. If a match fails, the following data is discarded. These registers can be read and written at anytime.

Address: Base address + 8h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	MA							
Reset	0	0	0	0	0	0	0	0

UARTx_MA1 field descriptions

Field	Description
7–0 MA	Match Address

49.3.10 UART Match Address Registers 2 (UARTx_MA2)

These registers can be read and written at anytime. The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated C4[MAEN] field is set. If a match occurs, the following data is transferred to the data register. If a match fails, the following data is discarded.

Address: Base address + 9h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	MA							
Reset	0	0	0	0	0	0	0	0

UARTx_MA2 field descriptions

Field	Description
7–0 MA	Match Address

49.3.11 UART Control Register 4 (UARTx_C4)

Address: Base address + Ah offset

Bit	7	6	5	4	3	2	1	0
Read / Write	MAEN1	MAEN2	M10	BRFA				
Reset	0	0	0	0	0	0	0	0

UARTx_C4 field descriptions

Field	Description
7 MAEN1	Match Address Mode Enable 1 See Match address operation for more information. 0 All data received is transferred to the data buffer if MAEN2 is cleared. 1 All data received with the most significant bit cleared, is discarded. All data received with the most significant bit set, is compared with contents of MA1 register. If no match occurs, the data is discarded. If match occurs, data is transferred to the data buffer. This field must be cleared when C7816[ISO7816E] is set/enabled.
6 MAEN2	Match Address Mode Enable 2 See Match address operation for more information. 0 All data received is transferred to the data buffer if MAEN1 is cleared. 1 All data received with the most significant bit cleared, is discarded. All data received with the most significant bit set, is compared with contents of MA2 register. If no match occurs, the data is discarded. If a match occurs, data is transferred to the data buffer. This field must be cleared when C7816[ISO7816E] is set/enabled.

Table continues on the next page...

UARTx_C4 field descriptions (continued)

Field	Description
5 M10	10-bit Mode select Causes a tenth, non-memory mapped bit to be part of the serial transmission. This tenth bit is generated and interpreted as a parity bit. The M10 field does not affect the LIN send or detect break behavior. If M10 is set, then both C1[M] and C1[PE] must also be set. This field must be cleared when C7816[ISO7816E] is set/enabled. See Data format (non ISO-7816) for more information. 0 The parity bit is the ninth bit in the serial transmission. 1 The parity bit is the tenth bit in the serial transmission.
4–0 BRFA	Baud Rate Fine Adjust This bit field is used to add more timing resolution to the average baud frequency, in increments of 1/32. See Baud rate generation for more information.

49.3.12 UART Control Register 5 (UARTx_C5)

Address: Base address + Bh offset

Bit	7	6	5	4	3	2	1	0
Read	TDMAS	0	RDMAS	0	0			
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_C5 field descriptions

Field	Description
7 TDMAS	Transmitter DMA Select Configures the transmit data register empty flag, S1[TDRE], to generate interrupt or DMA requests if C2[TIE] is set. **NOTE:** • If C2[TIE] is cleared, TDRE DMA and TDRE interrupt request signals are not asserted when the TDRE flag is set, regardless of the state of TDMAS. • If C2[TIE] and TDMAS are both set, then C2[TCIE] must be cleared, and D must not be written unless a DMA request is being serviced. 0 If C2[TIE] is set and the S1[TDRE] flag is set, the TDRE interrupt request signal is asserted to request interrupt service. 1 If C2[TIE] is set and the S1[TDRE] flag is set, the TDRE DMA request signal is asserted to request a DMA transfer.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 RDMAS	Receiver Full DMA Select Configures the receiver data register full flag, S1[RDRF], to generate interrupt or DMA requests if C2[RIE] is set. **NOTE:** If C2[RIE] is cleared, and S1[RDRF] is set, the RDRF DMA and RDFR interrupt request signals are not asserted, regardless of the state of RDMAS.

Table continues on the next page...

UARTx_C5 field descriptions (continued)

Field	Description
	0 If C2[RIE] and S1[RDRF] are set, the RDFR interrupt request signal is asserted to request an interrupt service. 1 If C2[RIE] and S1[RDRF] are set, the RDRF DMA request signal is asserted to request a DMA transfer.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

49.3.13 UART Extended Data Register (UARTx_ED)

This register contains additional information flags that are stored with a received dataword. This register may be read at any time but contains valid data only if there is a dataword in the receive FIFO.

NOTE

- The data contained in this register represents additional information regarding the conditions on which a dataword was received. The importance of this data varies with the application, and in some cases maybe completely optional. These fields automatically update to reflect the conditions of the next dataword whenever D is read.
- If S1[NF] and S1[PF] have not been set since the last time the receive buffer was empty, the NOISY and PARITYE fields will be zero.

Address: Base address + Ch offset

Bit	7	6	5	4	3	2	1	0
Read	NOISY	PARITYE	0					
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_ED field descriptions

Field	Description
7 NOISY	The current received dataword contained in D and C3[R8] was received with noise. 0 The dataword was received without noise. 1 The data was received with noise.
6 PARITYE	The current received dataword contained in D and C3[R8] was received with a parity error. 0 The dataword was received without a parity error. 1 The dataword was received with a parity error.

Table continues on the next page...

UARTx_ED field descriptions (continued)

Field	Description
5–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

49.3.14 UART Modem Register (UARTx_MODEM)

The MODEM register controls options for setting the modem configuration.

NOTE

RXRTSE, TXRTSPOL, TXRTSE, and TXCTSE must all be cleared when C7816[ISO7816EN] is enabled. This will cause the RTS to deassert during ISO-7816 wait times. The ISO-7816 protocol does not use the RTS and CTS signals.

Address: Base address + Dh offset

Bit	7	6	5	4	3	2	1	0
Read	0				RXRTSE	TXRTSPOL	TXRTSE	TXCTSE
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_MODEM field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 RXRTSE	Receiver request-to-send enable Allows the RTS output to control the CTS input of the transmitting device to prevent receiver overrun. **NOTE:** Do not set both RXRTSE and TXRTSE. 0 The receiver has no effect on RTS. 1 RTS is deasserted if the number of characters in the receiver data register (FIFO) is equal to or greater than RWFIFO[RXWATER]. RTS is asserted when the number of characters in the receiver data register (FIFO) is less than RWFIFO[RXWATER].
2 TXRTSPOL	Transmitter request-to-send polarity Controls the polarity of the transmitter RTS. TXRTSPOL does not affect the polarity of the receiver RTS. RTS will remain negated in the active low state unless TXRTSE is set. 0 Transmitter RTS is active low. 1 Transmitter RTS is active high.
1 TXRTSE	Transmitter request-to-send enable Controls RTS before and after a transmission.

Table continues on the next page...

UARTx_MODEM field descriptions (continued)

Field	Description
	0 The transmitter has no effect on RTS. 1 When a character is placed into an empty transmitter data buffer , RTS asserts one bit time before the start bit is transmitted. RTS deasserts one bit time after all characters in the transmitter data buffer and shift register are completely sent, including the last stop bit. (FIFO)(FIFO)
0 TXCTSE	Transmitter clear-to-send enable TXCTSE controls the operation of the transmitter. TXCTSE can be set independently from the state of TXRTSE and RXRTSE. 0 CTS has no effect on the transmitter. 1 Enables clear-to-send operation. The transmitter checks the state of CTS each time it is ready to send a character. If CTS is asserted, the character is sent. If CTS is deasserted, the signal TXD remains in the mark state and transmission is delayed until CTS is asserted. Changes in CTS as a character is being sent do not affect its transmission.

49.3.15 UART Infrared Register (UARTx_IR)

The IR register controls options for setting the infrared configuration.

Address: Base address + Eh offset

Bit	7	6	5	4	3	2	1	0
Read	0					IREN	TNP	
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_IR field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 IREN	Infrared enable Enables/disables the infrared modulation/demodulation. 0 IR disabled. 1 IR enabled.
1–0 TNP	Transmitter narrow pulse Enables whether the UART transmits a 1/16, 3/16, 1/32, or 1/4 narrow pulse. 00 3/16. 01 1/16. 10 1/32. 11 1/4.

49.3.16 UART FIFO Parameters (UARTx_PFIFO)

This register provides the ability for the programmer to turn on and off FIFO functionality. It also provides the size of the FIFO that has been implemented. This register may be read at any time. This register must be written only when C2[RE] and C2[TE] are cleared/not set and when the data buffer/FIFO is empty.

Address: Base address + 10h offset

Bit	7	6	5	4	3	2	1	0
Read	TXFE	TXFIFOSIZE			RXFE	RXFIFOSIZE		
Write								
Reset	0	*	*	*	0	*	*	*

* Notes:

- TXFIFOSIZE field: The reset value depends on whether the specific UART instance supports the FIFO and on the size of that FIFO. See the Chip Configuration details for more information on the FIFO size supported for each UART instance.
- RXFIFOSIZE field: The reset value depends on whether the specific UART instance supports the FIFO and on the size of that FIFO. See the Chip Configuration details for more information on the FIFO size supported for each UART instance.

UARTx_PFIFO field descriptions

Field	Description
7 TXFE	Transmit FIFO Enable When this field is set, the built in FIFO structure for the transmit buffer is enabled. The size of the FIFO structure is indicated by TXFIFOSIZE. If this field is not set, the transmit buffer operates as a FIFO of depth one dataword regardless of the value in TXFIFOSIZE. Both C2[TE] and C2[RE] must be cleared prior to changing this field. Additionally, TXFLUSH and RXFLUSH commands must be issued immediately after changing this field. 0 Transmit FIFO is not enabled. Buffer is depth 1. (Legacy support). 1 Transmit FIFO is enabled. Buffer is depth indicated by TXFIFOSIZE.
6–4 TXFIFOSIZE	Transmit FIFO. Buffer Depth The maximum number of transmit datawords that can be stored in the transmit buffer. This field is read only. 000 Transmit FIFO/Buffer depth = 1 dataword. 001 Transmit FIFO/Buffer depth = 4 datawords. 010 Transmit FIFO/Buffer depth = 8 datawords. 011 Transmit FIFO/Buffer depth = 16 datawords. 100 Transmit FIFO/Buffer depth = 32 datawords. 101 Transmit FIFO/Buffer depth = 64 datawords. 110 Transmit FIFO/Buffer depth = 128 datawords. 111 Reserved.
3 RXFE	Receive FIFO Enable

Table continues on the next page...

UARTx_PFIFO field descriptions (continued)

Field	Description
	When this field is set, the built in FIFO structure for the receive buffer is enabled. The size of the FIFO structure is indicated by the RXFIFOSIZE field. If this field is not set, the receive buffer operates as a FIFO of depth one dataword regardless of the value in RXFIFOSIZE. Both C2[TE] and C2[RE] must be cleared prior to changing this field. Additionally, TXFLUSH and RXFLUSH commands must be issued immediately after changing this field. 0 Receive FIFO is not enabled. Buffer is depth 1. (Legacy support) 1 Receive FIFO is enabled. Buffer is depth indicted by RXFIFOSIZE.
2–0 RXFIFOSIZE	Receive FIFO. Buffer Depth The maximum number of receive datawords that can be stored in the receive buffer before an overrun occurs. This field is read only. 000 Receive FIFO/Buffer depth = 1 dataword. 001 Receive FIFO/Buffer depth = 4 datawords. 010 Receive FIFO/Buffer depth = 8 datawords. 011 Receive FIFO/Buffer depth = 16 datawords. 100 Receive FIFO/Buffer depth = 32 datawords. 101 Receive FIFO/Buffer depth = 64 datawords. 110 Receive FIFO/Buffer depth = 128 datawords. 111 Reserved.

49.3.17 UART FIFO Control Register (UARTx_CFIFO)

This register provides the ability to program various control fields for FIFO operation. This register may be read or written at any time. Note that writing to TXFLUSH and RXFLUSH may result in data loss and requires careful action to prevent unintended/ unpredictable behavior. Therefore, it is recommended that TE and RE be cleared prior to flushing the corresponding FIFO.

Address: Base address + 11h offset

Bit	7	6	5	4	3	2	1	0
Read	0	0	0			RXOFE	TXOFE	RXUFE
Write	TXFLUSH	RXFLUSH						
Reset	0	0	0	0	0	0	0	0

UARTx_CFIFO field descriptions

Field	Description
7 TXFLUSH	Transmit FIFO/Buffer Flush Writing to this field causes all data that is stored in the transmit FIFO/buffer to be flushed. This does not affect data that is in the transmit shift register. 0 No flush operation occurs. 1 All data in the transmit FIFO/Buffer is cleared out.

Table continues on the next page...

UARTx_CFIFO field descriptions (continued)

Field	Description
6 RXFLUSH	Receive FIFO/Buffer Flush Writing to this field causes all data that is stored in the receive FIFO/buffer to be flushed. This does not affect data that is in the receive shift register. 0 No flush operation occurs. 1 All data in the receive FIFO/buffer is cleared out.
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 RXOFE	Receive FIFO Overflow Interrupt Enable When this field is set, the RXOF flag generates an interrupt to the host. 0 RXOF flag does not generate an interrupt to the host. 1 RXOF flag generates an interrupt to the host.
1 TXOFE	Transmit FIFO Overflow Interrupt Enable When this field is set, the TXOF flag generates an interrupt to the host. 0 TXOF flag does not generate an interrupt to the host. 1 TXOF flag generates an interrupt to the host.
0 RXUFE	Receive FIFO Underflow Interrupt Enable When this field is set, the RXUF flag generates an interrupt to the host. 0 RXUF flag does not generate an interrupt to the host. 1 RXUF flag generates an interrupt to the host.

49.3.18 UART FIFO Status Register (UARTx_SFIFO)

This register provides status information regarding the transmit and receiver buffers/ FIFOs, including interrupt information. This register may be written to or read at any time.

Address: Base address + 12h offset

Bit	7	6	5	4	3	2	1	0
Read	TXEMPT	RXEMPT	0			RXOF	TXOF	RXUF
Write						w1c	w1c	w1c
Reset	1	1	0	0	0	0	0	0

UARTx_SFIFO field descriptions

Field	Description
7 TXEMPT	Transmit Buffer/FIFO Empty Asserts when there is no data in the Transmit FIFO/buffer. This field does not take into account data that is in the transmit shift register.

Table continues on the next page...

UARTx_SFIFO field descriptions (continued)

Field	Description
	0 Transmit buffer is not empty. 1 Transmit buffer is empty.
6 RXEMPT	Receive Buffer/FIFO Empty Asserts when there is no data in the receive FIFO/Buffer. This field does not take into account data that is in the receive shift register. 0 Receive buffer is not empty. 1 Receive buffer is empty.
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 RXOF	Receiver Buffer Overflow Flag Indicates that more data has been written to the receive buffer than it can hold. This field will assert regardless of the value of CFIFO[RXOFE]. However, an interrupt will be issued to the host only if CFIFO[RXOFE] is set. This flag is cleared by writing a 1. 0 No receive buffer overflow has occurred since the last time the flag was cleared. 1 At least one receive buffer overflow has occurred since the last time the flag was cleared.
1 TXOF	Transmitter Buffer Overflow Flag Indicates that more data has been written to the transmit buffer than it can hold. This field will assert regardless of the value of CFIFO[TXOFE]. However, an interrupt will be issued to the host only if CFIFO[TXOFE] is set. This flag is cleared by writing a 1. 0 No transmit buffer overflow has occurred since the last time the flag was cleared. 1 At least one transmit buffer overflow has occurred since the last time the flag was cleared.
0 RXUF	Receiver Buffer Underflow Flag Indicates that more data has been read from the receive buffer than was present. This field will assert regardless of the value of CFIFO[RXUFE]. However, an interrupt will be issued to the host only if CFIFO[RXUFE] is set. This flag is cleared by writing a 1. 0 No receive buffer underflow has occurred since the last time the flag was cleared. 1 At least one receive buffer underflow has occurred since the last time the flag was cleared.

49.3.19 UART FIFO Transmit Watermark (UARTx_TWFIFO)

This register provides the ability to set a programmable threshold for notification of needing additional transmit data. This register may be read at any time but must be written only when C2[TE] is not set. Changing the value of the watermark will not clear the S1[TDRE] flag.

Address: Base address + 13h offset

Bit	7	6	5	4	3	2	1	0
Read Write	TXWATER							
Reset	0	0	0	0	0	0	0	0

UARTx_TWFIFO field descriptions

Field	Description
7–0 TXWATER	Transmit Watermark When the number of datawords in the transmit FIFO/buffer is equal to or less than the value in this register field, an interrupt via S1[TDRE] or a DMA request via C5[TDMAS] is generated as determined by C5[TDMAS] and C2[TIE]. For proper operation, the value in TXWATER must be set to be less than the size of the transmit buffer/FIFO size as indicated by PFIFO[TXFIFOSIZE] and PFIFO[TXFE].

49.3.20 UART FIFO Transmit Count (UARTx_TCFIFO)

This is a read only register that indicates how many datawords are currently in the transmit buffer/FIFO. It may be read at any time.

Address: Base address + 14h offset

Bit	7	6	5	4	3	2	1	0
Read	TXCOUNT							
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_TCFIFO field descriptions

Field	Description
7–0 TXCOUNT	Transmit Counter The value in this register indicates the number of datawords that are in the transmit FIFO/buffer. If a dataword is being transmitted, that is, in the transmit shift register, it is not included in the count. This value may be used in conjunction with PFIFO[TXFIFOSIZE] to calculate how much room is left in the transmit FIFO/buffer.

49.3.21 UART FIFO Receive Watermark (UARTx_RWFIFO)

This register provides the ability to set a programmable threshold for notification of the need to remove data from the receiver FIFO/buffer. This register may be read at any time but must be written only when C2[RE] is not asserted. Changing the value in this register will not clear S1[RDRF].

Address: Base address + 15h offset

Bit	7	6	5	4	3	2	1	0
Read Write	RXWATER							
Reset	0	0	0	0	0	0	0	1

UARTx_RWFIFO field descriptions

Field	Description
7–0 RXWATER	Receive Watermark When the number of datawords in the receive FIFO/buffer is equal to or greater than the value in this register field, an interrupt via S1[RDRF] or a DMA request via C5[RDMAS] is generated as determined by C5[RDMAS] and C2[RIE]. For proper operation, the value in RXWATER must be set to be less than the receive FIFO/buffer size as indicated by PFIFO[RXFIFOSIZE] and PFIFO[RXFE] and must be greater than 0.

49.3.22 UART FIFO Receive Count (UARTx_RCFIFO)

This is a read only register that indicates how many datawords are currently in the receive FIFO/buffer. It may be read at any time.

Address: Base address + 16h offset

Bit	7	6	5	4	3	2	1	0
Read	RXCOUNT							
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_RCFIFO field descriptions

Field	Description
7–0 RXCOUNT	Receive Counter The value in this register indicates the number of datawords that are in the receive FIFO/buffer. If a dataword is being received, that is, in the receive shift register, it is not included in the count. This value may be used in conjunction with PFIFO[RXFIFOSIZE] to calculate how much room is left in the receive FIFO/buffer.

49.3.23 UART 7816 Control Register (UARTx_C7816)

The C7816 register is the primary control register for ISO-7816 specific functionality. This register is specific to 7816 functionality and the values in this register have no effect on UART operation and should be ignored if ISO_7816E is not set/enabled. This register may be read at any time but values must be changed only when ISO_7816E is not set.

Address: Base address + 18h offset

Bit	7	6	5	4	3	2	1	0
Read	0			ONACK	ANACK	INIT	TTYPE	ISO_7816E
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_C7816 field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 ONACK	Generate NACK on Overflow When this field is set, the receiver automatically generates a NACK response if a receive buffer overrun occurs, as indicated by S1[OR]. In many systems, this results in the transmitter resending the packet that overflowed until the retransmit threshold for that transmitter is reached. A NACK is generated only if TTYPE=0. This field operates independently of ANACK. See . Overrun NACK considerations 0 The received data does not generate a NACK when the receipt of the data results in an overflow event. 1 If the receiver buffer overflows, a NACK is automatically sent on a received character.
3 ANACK	Generate NACK on Error When this field is set, the receiver automatically generates a NACK response if a parity error occurs or if INIT is set and an invalid initial character is detected. A NACK is generated only if TTYPE = 0. If ANACK is set, the UART attempts to retransmit the data indefinitely. To stop retransmission attempts, clear C2[TE] or ISO_7816E and do not set until S1[TC] sets C2[TE] again. 0 No NACK is automatically generated. 1 A NACK is automatically generated if a parity error is detected or if an invalid initial character is detected.
2 INIT	Detect Initial Character When this field is set, all received characters are searched for a valid initial character. If an invalid initial character is identified, and ANACK is set, a NACK is sent. All received data is discarded and error flags blocked (S1[NF], S1[OR], S1[FE], S1[PF], IS7816[WT], IS7816[CWT], IS7816[BWT], IS7816[GTV]) until a valid initial character is detected. Upon detecting a valid initial character, the configuration values S2[MSBF], C3[TXINV], and S2[RXINV] are automatically updated to reflect the initial character that was received. The actual INIT data value is not stored in the receive buffer. Additionally, upon detection of a valid initial character, IS7816[INITD] is set and an interrupt issued as programmed by IE7816[INITDE]. When a valid initial character is detected, INIT is automatically cleared. This Initial Character Detect feature is supported only in T = 0 protocol mode. 0 Normal operating mode. Receiver does not seek to identify initial character. 1 Receiver searches for initial character.
1 TTYPE	Transfer Type Indicates the transfer protocol being used. See ISO-7816 / smartcard support for more details. 0 T = 0 per the ISO-7816 specification. 1 T = 1 per the ISO-7816 specification.
0 ISO_7816E	ISO-7816 Functionality Enabled Indicates that the UART is operating according to the ISO-7816 protocol. **NOTE:** This field must be modified only when no transmit or receive is occurring. If this field is changed during a data transfer, the data being transmitted or received may be transferred incorrectly. 0 ISO-7816 functionality is turned off/not enabled. 1 ISO-7816 functionality is turned on/enabled.

49.3.24 UART 7816 Interrupt Enable Register (UARTx_IE7816)

The IE7816 register controls which flags result in an interrupt being issued. This register is specific to 7816 functionality, the corresponding flags that drive the interrupts are not asserted when 7816E is not set/enabled. However, these flags may remain set if they are asserted while 7816E was set and not subsequently cleared. This register may be read or written to at any time.

Address: Base address + 19h offset

Bit	7	6	5	4	3	2	1	0
Read	WTE	CWTE	BWTE	INITDE	0	GTVE	TXTE	RXTE
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_IE7816 field descriptions

Field	Description
7 WTE	Wait Timer Interrupt Enable 0 The assertion of IS7816[WT] does not result in the generation of an interrupt. 1 The assertion of IS7816[WT] results in the generation of an interrupt.
6 CWTE	Character Wait Timer Interrupt Enable 0 The assertion of IS7816[CWT] does not result in the generation of an interrupt. 1 The assertion of IS7816[CWT] results in the generation of an interrupt.
5 BWTE	Block Wait Timer Interrupt Enable 0 The assertion of IS7816[BWT] does not result in the generation of an interrupt. 1 The assertion of IS7816[BWT] results in the generation of an interrupt.
4 INITDE	Initial Character Detected Interrupt Enable 0 The assertion of IS7816[INITD] does not result in the generation of an interrupt. 1 The assertion of IS7816[INITD] results in the generation of an interrupt.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 GTVE	Guard Timer Violated Interrupt Enable 0 The assertion of IS7816[GTV] does not result in the generation of an interrupt. 1 The assertion of IS7816[GTV] results in the generation of an interrupt.
1 TXTE	Transmit Threshold Exceeded Interrupt Enable 0 The assertion of IS7816[TXT] does not result in the generation of an interrupt. 1 The assertion of IS7816[TXT] results in the generation of an interrupt.
0 RXTE	Receive Threshold Exceeded Interrupt Enable 0 The assertion of IS7816[RXT] does not result in the generation of an interrupt. 1 The assertion of IS7816[RXT] results in the generation of an interrupt.

49.3.25 UART 7816 Interrupt Status Register (UARTx_IS7816)

The IS7816 register provides a mechanism to read and clear the interrupt flags. All flags/interrupts are cleared by writing a 1 to the field location. Writing a 0 has no effect. All bits are "sticky", meaning they indicate that only the flag condition that occurred since the last time the bit was cleared, not that the condition currently exists. The status flags are set regardless of whether the corresponding field in the IE7816 is set or cleared. The IE7816 controls only if an interrupt is issued to the host processor. This register is specific to 7816 functionality and the values in this register have no affect on UART operation and should be ignored if 7816E is not set/enabled. This register may be read or written at anytime.

Address: Base address + 1Ah offset

Bit	7	6	5	4	3	2	1	0
Read	WT	CWT	BWT	INITD	0	GTV	TXT	RXT
Write	w1c	w1c	w1c	w1c		w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

UARTx_IS7816 field descriptions

Field	Description
7 WT	Wait Timer Interrupt Indicates that the wait time, the time between the leading edge of a character being transmitted and the leading edge of the next response character, has exceeded the programmed value. This flag asserts only when C7816[TTYPE] = 0. This interrupt is cleared by writing 1. 0 Wait time (WT) has not been violated. 1 Wait time (WT) has been violated.
6 CWT	Character Wait Timer Interrupt Indicates that the character wait time, the time between the leading edges of two consecutive characters in a block, has exceeded the programmed value. This flag asserts only when C7816[TTYPE] = 1. This interrupt is cleared by writing 1. 0 Character wait time (CWT) has not been violated. 1 Character wait time (CWT) has been violated.
5 BWT	Block Wait Timer Interrupt Indicates that the block wait time, the time between the leading edge of first received character of a block and the leading edge of the last character the previously transmitted block, has exceeded the programmed value. This flag asserts only when C7816[TTYPE] = 1.This interrupt is cleared by writing 1. 0 Block wait time (BWT) has not been violated. 1 Block wait time (BWT) has been violated.
4 INITD	Initial Character Detected Interrupt Indicates that a valid initial character is received. This interrupt is cleared by writing 1.

Table continues on the next page...

UARTx_IS7816 field descriptions (continued)

Field	Description
	0 A valid initial character has not been received. 1 A valid initial character has been received.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 GTV	Guard Timer Violated Interrupt Indicates that one or more of the character guard time, block guard time, or guard time are violated. This interrupt is cleared by writing 1. 0 A guard time (GT, CGT, or BGT) has not been violated. 1 A guard time (GT, CGT, or BGT) has been violated.
1 TXT	Transmit Threshold Exceeded Interrupt Indicates that the transmit NACK threshold has been exceeded as indicated by ET7816[TXTHRESHOLD]. Regardless of whether this flag is set, the UART continues to retransmit indefinitely. This flag asserts only when C7816[TTYPE] = 0. If 7816E is cleared/disabled, ANACK is cleared/disabled, C2[TE] is cleared/disabled, C7816[TTYPE] = 1, or packet is transferred without receiving a NACK, the internal NACK detection counter is cleared and the count restarts from zero on the next received NACK. This interrupt is cleared by writing 1. 0 The number of retries and corresponding NACKS does not exceed the value in ET7816[TXTHRESHOLD]. 1 The number of retries and corresponding NACKS exceeds the value in ET7816[TXTHRESHOLD].
0 RXT	Receive Threshold Exceeded Interrupt Indicates that there are more than ET7816[RXTHRESHOLD] consecutive NACKS generated in response to parity errors on received data. This flag requires ANACK to be set. Additionally, this flag asserts only when C7816[TTYPE] = 0. Clearing this field also resets the counter keeping track of consecutive NACKS. The UART will continue to attempt to receive data regardless of whether this flag is set. If 7816E is cleared/disabled, RE is cleared/disabled, C7816[TTYPE] = 1, or packet is received without needing to issue a NACK, the internal NACK detection counter is cleared and the count restarts from zero on the next transmitted NACK. This interrupt is cleared by writing 1. 0 The number of consecutive NACKS generated as a result of parity errors and buffer overruns is less than or equal to the value in ET7816[RXTHRESHOLD]. 1 The number of consecutive NACKS generated as a result of parity errors and buffer overruns is greater than the value in ET7816[RXTHRESHOLD].

49.3.26 UART 7816 Wait Parameter Register (UARTx_WP7816T0)

The WP7816 register contains constants used in the generation of various wait timer counters. To save register space, this register is used differently when C7816[TTYPE] = 0 and C7816[TTYPE] = 1. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Bh offset

Bit	7	6	5	4	3	2	1	0
Read Write	WI							
Reset	0	0	0	0	1	0	1	0

UARTx_WP7816T0 field descriptions

Field	Description
7–0 WI	Wait Time Integer (C7816[TTYPE] = 0) Used to calculate the value used for the WT counter. It represents a value between 1 and 255. The value of zero is not valid. This value is used only when C7816[TTYPE] = 0. See Wait time and guard time parameters.

49.3.27 UART 7816 Wait Parameter Register (UARTx_WP7816T1)

The WP7816 register contains constants used in the generation of various wait timer counters. To save register space, this register is used differently when C7816[TTYPE] = 0 and C7816[TTYPE] = 1. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Bh offset

Bit	7	6	5	4	3	2	1	0
Read Write	CWI				BWI			
Reset	0	0	0	0	1	0	1	0

UARTx_WP7816T1 field descriptions

Field	Description
7–4 CWI	Character Wait Time Integer (C7816[TTYPE] = 1) Used to calculate the value used for the CWT counter. It represents a value between 0 and 15. This value is used only when C7816[TTYPE] = 1. See Wait time and guard time parameters .
3–0 BWI	Block Wait Time Integer(C7816[TTYPE] = 1) Used to calculate the value used for the BWT counter. It represent a value between 0 and 15. This value is used only when C7816[TTYPE] = 1. See Wait time and guard time parameters .

49.3.28 UART 7816 Wait N Register (UARTx_WN7816)

The WN7816 register contains a parameter that is used in the calculation of the guard time counter. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Ch offset

Bit	7	6	5	4	3	2	1	0
Read Write	GTN							
Reset	0	0	0	0	0	0	0	0

UARTx_WN7816 field descriptions

Field	Description
7–0 GTN	Guard Band N Defines a parameter used in the calculation of GT, CGT, and BGT counters. The value represents an integer number between 0 and 255. See Wait time and guard time parameters .

49.3.29 UART 7816 Wait FD Register (UARTx_WF7816)

The WF7816 contains parameters that are used in the generation of various counters including GT, CGT, BGT, WT, and BWT. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Dh offset

Bit	7	6	5	4	3	2	1	0
Read Write	GTFD							
Reset	0	0	0	0	0	0	0	1

UARTx_WF7816 field descriptions

Field	Description
7–0 GTFD	FD Multiplier Used as another multiplier in the calculation of WT and BWT. This value represents a number between 1 and 255. The value of 0 is invalid. This value is not used in baud rate generation. See Wait time and guard time parameters and Baud rate generation .

49.3.30 UART 7816 Error Threshold Register (UARTx_ET7816)

The ET7816 register contains fields that determine the number of NACKs that must be received or transmitted before the host processor is notified. This register may be read at anytime. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Eh offset

Bit	7	6	5	4	3	2	1	0
Read Write	TXTHRESHOLD				RXTHRESHOLD			
Reset	0	0	0	0	0	0	0	0

UARTx_ET7816 field descriptions

Field	Description
7–4 TXTHRESHOLD	Transmit NACK Threshold The value written to this field indicates the maximum number of failed attempts (NACKs) a transmitted character can have before the host processor is notified. This field is meaningful only when

Table continues on the next page...

UARTx_ET7816 field descriptions (continued)

Field	Description
	C7816[TTYPE] = 0 and C7816[ANACK] = 1. The value read from this field represents the number of consecutive NACKs that have been received since the last successful transmission. This counter saturates at 4'hF and does not wrap around. Regardless of how many NACKs that are received, the UART continues to retransmit indefinitely. This flag only asserts when C7816[TTYPE] = 0. For additional information see the IS7816[TXT] field description. 0 TXT asserts on the first NACK that is received. 1 TXT asserts on the second NACK that is received.
3–0 RXTHRESHOLD	Receive NACK Threshold The value written to this field indicates the maximum number of consecutive NACKs generated as a result of a parity error or receiver buffer overruns before the host processor is notified. After the counter exceeds that value in the field, the IS7816[RXT] is asserted. This field is meaningful only when C7816[TTYPE] = 0. The value read from this field represents the number of consecutive NACKs that have been transmitted since the last successful reception. This counter saturates at 4'hF and does not wrap around. Regardless of the number of NACKs sent, the UART continues to receive valid packets indefinitely. For additional information, see IS7816[RXT] field description.

49.3.31 UART 7816 Transmit Length Register (UARTx_TL7816)

The TL7816 register is used to indicate the number of characters contained in the block being transmitted. This register is used only when C7816[TTYPE] = 1. This register may be read at anytime. This register must be written only when C2[TE] is not enabled.

Address: Base address + 1Fh offset

Bit	7	6	5	4	3	2	1	0
Read Write	TLEN							
Reset	0	0	0	0	0	0	0	0

UARTx_TL7816 field descriptions

Field	Description
7–0 TLEN	Transmit Length This value plus four indicates the number of characters contained in the block being transmitted. This register is automatically decremented by 1 for each character in the information field portion of the block. Additionally, this register is automatically decremented by 1 for the first character of a CRC in the epilogue field. Therefore, this register must be programmed with the number of bytes in the data packet if an LRC is being transmitted, and the number of bytes + 1 if a CRC is being transmitted. This register is not decremented for characters that are assumed to be part of the Prologue field, that is, the first three characters transmitted in a block, or the LRC or last CRC character in the Epilogue field, that is, the last character transmitted. This field must be programed or adjusted only when C2[TE] is cleared.

49.3.32 UART CEA709.1-B Control Register 6 (UARTx_C6)

Address: Base address + 21h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	EN709	TX709	CE	CP	0			
Reset	0	0	0	0	0	0	0	0

UARTx_C6 field descriptions

Field	Description
7 EN709	EN709 Enables the CEA709.1-B feature. 0 CEA709.1-B is disabled. 1 CEA709.1-B is enabled
6 TX709	CEA709.1-B Transmit Enable Starts CEA709.1-B transmission. This bit is automatically cleared after completely transmitting the packet 0 CEA709.1-B transmitter is disabled. 1 CEA709.1-B transmitter is enabled.
5 CE	Collision Enable Enables the collision detect functionality. 0 Collision detect feature is disabled. 1 Collision detect feature is enabled.
4 CP	Collision Signal Polarity Indicates the polarity of the collision signal. 0 Collision signal is active low. 1 Collision signal is active high.
3–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

49.3.33 UART CEA709.1-B Packet Cycle Time Counter High (UARTx_PCTH)

Address: Base address + 22h offset

Bit	7	6	5	4	3	2	1	0
Read / Write	PCTH							
Reset	0	0	0	0	0	0	0	0

UARTx_PCTH field descriptions

Field	Description
7–0 PCTH	Packet Cycle Time Counter High Indicates the most significant byte of maximum period after the line code violation (B1 and SDT expired) for which the bus could remain idle without decrementing back log count. If the time elapsed after line code violation (B1 and SDT expired) is greater than packet cycle time, then packet cycle timer expired interrupt is generated. It is measured in terms of bit times, that is, the time it takes for a single bit or one differential Manchester symbol to be transmitted. This is medium dependent and hence does not usually require adjustment and is programmed only once.

49.3.34 UART CEA709.1-B Packet Cycle Time Counter Low (UARTx_PCTL)

Address: Base address + 23h offset

Bit	7	6	5	4	3	2	1	0
Read Write	PCTL							
Reset	0	0	0	0	0	0	0	0

UARTx_PCTL field descriptions

Field	Description
7–0 PCTL	Packet Cycle Time Counter Low Indicates the least significant byte of maximum period after the line code violation (B1 and SDT expired) for which the bus could remain idle without decrementing back log count. If the time elapsed after line code violation (B1 and SDT expired) is greater than packet cycle time, then packet cycle timer expired interrupt is generated. It is measured in terms of bit times, that is, the time it takes for a single bit or one Differential Manchester symbol to be transmitted. This is medium dependent and therefore does not usually require adjustment and is programmed only once.

49.3.35 UART CEA709.1-B Beta1 Timer (UARTx_B1T)

Address: Base address + 24h offset

Bit	7	6	5	4	3	2	1	0
Read Write	B1T							
Reset	0	0	0	0	0	0	0	0

UARTx_B1T field descriptions

Field	Description
7–0 B1T	Beta1 Timer Beta1 delay is a value that is system dependent and usually does not require adjustment. It is programmed only once and measured in bit times.

49.3.36 UART CEA709.1-B Secondary Delay Timer High (UARTx_SDTH)

Address: Base address + 25h offset

Bit	7	6	5	4	3	2	1	0
Read Write	SDTH							
Reset	0	0	0	0	0	0	0	0

UARTx_SDTH field descriptions

Field	Description
7–0 SDTH	Secondary Delay Timer High This is the most significant byte of the secondary delay timer and is set by software. This is generally a variable value that must be set for each data message to be transmitted. It is measured in bit times, that is, the time that it takes for a single bit or one differential Manchester symbol to be transmitted. This value must be between 0 and (BL*Wbase) + (ProritySlots -1), Beta2 timeslots. A value of zero indicates that the queued packet will be sent immediately upon expiration of the beta1 timer.

49.3.37 UART CEA709.1-B Secondary Delay Timer Low (UARTx_SDTL)

Address: Base address + 26h offset

Bit	7	6	5	4	3	2	1	0
Read Write	SDTL							
Reset	0	0	0	0	0	0	0	0

UARTx_SDTL field descriptions

Field	Description
7–0 SDTL	Secondary Delay Timer Low This is the least significant byte of the secondary delay timer and is set by software. This is generally a variable value that must be set for each data message to be transmitted. It is measured in bit times, that is, the time that it takes for a single bit or one Differential Manchester symbol to be transmitted. This value must be between 0 and (BL*Wbase) + (ProritySlots -1), Beta2 timeslots. A value of zero indicates that the queued packet will be sent immediately upon expiration of the Beta1 timer.

49.3.38 UART CEA709.1-B Preamble (UARTx_PRE)

Address: Base address + 27h offset

Bit	7	6	5	4	3	2	1	0
Read Write	PREAMBLE							
Reset	0	0	0	0	0	0	0	0

UARTx_PRE field descriptions

Field	Description
7–0 PREAMBLE	CEA709.1-B Preamble Register The number of bit-sync characters that occur prior to the byte-sync character when preamble is transmitted. **NOTE:** The minimum preamble length supported by twisted pair wire is four bit-sync fields.

49.3.39 UART CEA709.1-B Transmit Packet Length (UARTx_TPL)

Address: Base address + 28h offset

Bit	7	6	5	4	3	2	1	0
Read Write	TPL							
Reset	0	0	0	0	0	0	0	0

UARTx_TPL field descriptions

Field	Description
7–0 TPL	Transmit Packet Length Register Length of the data packet in bytes that is transmitted by CEA709.1-B transmitter. This includes the CRC packet as well.

49.3.40 UART CEA709.1-B Interrupt Enable Register (UARTx_IE)

Address: Base address + 29h offset

Bit	7	6	5	4	3	2	1	0
Read	0	WBEIE	ISDIE	PRXIE	PTXIE	PCTEIE	PSIE	TXFIE
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_IE field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 WBEIE	WBASE Expired Interrupt Enable Interrupt enable for WBASE expired flag. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 ISDIE	Initial Sync Detection Interrupt Enable Interrupt enable for initial synchronization detection flag.

Table continues on the next page...

UARTx_IE field descriptions (continued)

Field	Description
	NOTE: This field cannot be cleared except by disabling CEA709. Therefore, ISDIE must be cleared when the first initial sync detection interrupt occurs. If the ISD interrupt is not disabled in the interrupt handler, then user will continuously get interrupts. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 PRXIE	Packet Received Interrupt Enable Interrupt enable for packet received flag. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 PTXIE	Packet Transmitted Interrupt Enable Interrupt enable for packet transmitted flag. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 PCTEIE	Packet Cycle Timer Interrupt Enable Interrupt enable for packet cycle time expired flag. 0 Interrupt is disabled. 1 Interrupt is enabled.
1 PSIE	Preamble Start Interrupt Enable Interrupt enable for preamble start flag. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 TXFIE	Transmission Fail Interrupt Enable Interrupt enable for transmission fail flag. 0 Interrupt is disabled. 1 Interrupt is enabled.

49.3.41 UART CEA709.1-B WBASE (UARTx_WB)

Address: Base address + 2Ah offset

Bit	7	6	5	4	3	2	1	0
Read Write	WBASE							
Reset	0	0	0	0	0	0	0	0

UARTx_WB field descriptions

Field	Description
7–0 WBASE	CEA709.1-B WBASE register

UARTx_WB field descriptions (continued)

Field	Description
	Size of the basic randomizing window in bit periods after Beta1 time period. It is measured in bit times.

49.3.42 UART CEA709.1-B Status Register (UARTx_S3)

Address: Base address + 2Bh offset

Bit	7	6	5	4	3	2	1	0
Read	PEF	WBEF	ISD	PRXF	PTXF	PCTEF	PSF	TXFF
Write	w1c	w1c		w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

UARTx_S3 field descriptions

Field	Description
7 PEF	Preamble Error Flag Indicates that the received preamble has an error. If the received preamble length is greater than or less than the transmit preamble length, the preamble error flag is asserted. This flag is cleared by writing 1. 0 Preamble is correct. 1 Preamble has an error.
6 WBEF	Wbase Expired Flag Indicates that the Wbase time period has expired after beta1 time slots. This flag is cleared by writing 1. 0 WBASE time period has not expired. 1 WBASE time period has expired after beta1 time slots.
5 ISD	Initial Sync Detect Indicates that initially, a valid one and a line code violation is detected. This flag is cleared by deasserting EN709 bit. 0 Initial sync is not detected. 1 Initial sync is detected.
4 PRXF	Packet Received Flag Indicates that complete packet is received. This flag is cleared by writing 1. 0 Packet is not received. 1 Packet is received.
3 PTXF	Packet Transmitted Flag Indicates that complete packet is transmitted. This flag is cleared by writing 1. In case TX packet gets aborted due to FIFO becoming empty or an overflow, packet transmitted flag will still be generated. 0 Packet transmission is not complete. 1 Packet transmission is complete.

Table continues on the next page...

UARTx_S3 field descriptions (continued)

Field	Description
2 PCTEF	Packet Cycle Timer Expired Flag Indicates that packet cycle time period has expired with no activity on the line. This flag is cleared by writing 1. 0 Packet cycle time has not expired. 1 Packet cycle time has expired.
1 PSF	Preamble Start Flag Indicates start of the preamble while the packet is being transmitted. This flag is cleared by writing 1. 0 Preamble start is not detected. 1 Preamble start is detected.
0 TXFF	Transmission Fail Flag This flag indicates that transmission could not proceed. This flag is asserted when the packet is queued for transmission but before the random delay is expired an incoming receive packet is detected. This flag is also asserted while transmission when the TX fifo becomes empty or overflows. During these cases Line Code Violation is transmitted on TX line immediately after current byte or preamble transmission is finished, without waiting for completion of transmit packet length. If the transmission fail flag is asserted then TX709 bit of UART_C6 register is cleared. This flag is cleared by writing `1'. 0 Transmission continues normally. 1 Transmission has failed.

49.3.43 UART CEA709.1-B Status Register (UARTx_S4)

Address: Base address + 2Ch offset

Bit	7	6	5	4	3	2	1	0
Read	0			INITF	CDET		ILCV	FE
Write					w1c		w1c	w1c
Reset	0	0	0	0	0	0	0	0

UARTx_S4 field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 INITF	Initial Synchronization Fail Flag Indicates that the initial synchronization has failed and the packet cycle time has expired after enabling EN709 register. This flag is cleared if EN709 is cleared. 0 Initial synchronization has not failed. 1 Initial synchronization has failed.
3–2 CDET	CDET

Table continues on the next page...

UARTx_S4 field descriptions (continued)

Field	Description
	Indicates when the collision occurs during transmission. This flag is cleared by writing 2'b11. If the collision flag is not cleared by software and a valid collision pulse is detected during some other phase of transmission, then collision flag continues to indicate the previous value. 00 No collision. 01 Collision occurred during preamble. 10 Collision occurred during byte sync or data. 11 Collision occurred during line code violation.
1 ILCV	Improper Line Code Violation Indicates that line code violation received is not proper. This flag is cleared by writing 1. 0 Line code violation received is proper. 1 Line code violation received is improper, that is, less than three bit periods.
0 FE	Framing Error Indicates that the received CEA709.1-B packet has not finished at a byte boundary. This flag is cleared by writing 1. 0 Received packet is byte aligned. 1 Received packet is not byte aligned.

49.3.44 UART CEA709.1-B Received Packet Length (UARTx_RPL)

Address: Base address + 2Dh offset

Bit	7	6	5	4	3	2	1	0
Read	RPL							
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_RPL field descriptions

Field	Description
7–0 RPL	Received Packet Length Indicates the length of received packet in bytes. If the received packet is not byte aligned, the partial byte received is appended by zeros.

49.3.45 UART CEA709.1-B Received Preamble Length (UARTx_RPREL)

Address: Base address + 2Eh offset

Bit	7	6	5	4	3	2	1	0
Read	RPREL							
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_RPREL field descriptions

Field	Description
7–0 RPREL	Received Preamble Length Indicates the number of bit sync fields received in the preamble. The length saturates to 255 for the cases where the preamble (bit sync) is larger than 255. **NOTE:** The preamble length counter resets whenever there is no edge transition (SSSSS or DDDDD) during the resynchronization phase of preamble detection.

49.3.46 UART CEA709.1-B Collision Pulse Width (UARTx_CPW)

Address: Base address + 2Fh offset

Bit	7	6	5	4	3	2	1	0
Read Write	CPW							
Reset	0	0	0	0	0	0	0	0

UARTx_CPW field descriptions

Field	Description
7–0 CPW	CEA709.1-B CPW register Indicates the width of valid collision pulse in terms of IPG clock cycles.

49.3.47 UART CEA709.1-B Receive Indeterminate Time (UARTx_RIDT)

Address: Base address + 30h offset

Bit	7	6	5	4	3	2	1	0
Read Write	RIDT							
Reset	0	0	0	0	0	0	0	0

UARTx_RIDT field descriptions

Field	Description
7–0 RIDT	CEA709.1-B Receive IDT register Indicates the indeterminate time period measured in bit times after reception during which any activity on RX line will be discarded. Indeterminate time period value should be less than Beta1 timer value.

49.3.48 UART CEA709.1-B Transmit Indeterminate Time (UARTx_TIDT)

Address: Base address + 31h offset

Bit	7	6	5	4	3	2	1	0
Read Write	TIDT							
Reset	0	0	0	0	0	0	0	0

UARTx_TIDT field descriptions

Field	Description
7–0 TIDT	CEA709.1-B Transmit IDT Register This register indicates the indeterminate time period measured in bit times after transmission during which any activity on TX line will be discarded. Indeterminate time period value should be less than Beta1 timer value.

49.4 Functional description

This section provides a complete functional description of the UART block.

The UART allows full duplex, asynchronous, NRZ serial communication between the CPU and remote devices, including other CPUs. The UART transmitter and receiver operate independently, although they use the same baud rate generator. The CPU monitors the status of the UART, writes the data to be transmitted, and processes received data.

49.4.1 CEA709.1-B

The UART provides support for CEA709.1-B, which is commonly used in building automation, home networking, including all key building automation subsystems such as heating, ventilating, air-conditioning, lighting, security, fire detection, access control, and energy monitoring.

49.4.1.1 CEA709.1-B packet cycle

The following figure illustrates the frame format and Differential Manchester encoding. Differential Manchester encoding requires that each transmitted bit includes a clock transition at the start of the bit period. This allows synchronization with the receiver.

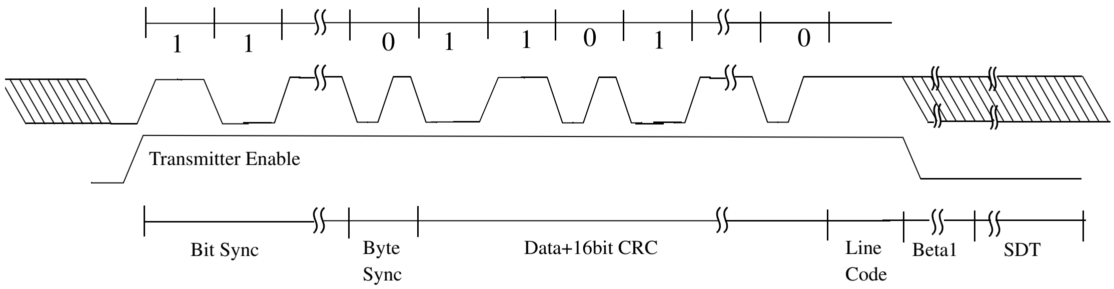


Figure 49-337. Frame format with differential manchester encoding

A logic zero is indicated with the presence of a transition in the middle of the bit period and a logic one is indicated by the absence of any transition. When transitions occur at the start of the bit time, polarity is arbitrary because the last bit of a transmission has no trailing clock edge. A transmitter will transmit a preamble at the beginning of a packet to allow other nodes to synchronize their receiver clocks. The preamble comprises a bit-sync field followed by a byte-sync field. The bit-sync field is a series of differential Manchester logic ones and the byte-sync field is a single differential Manchester logic zero. The byte-sync field marks the end of the preamble and the start of the data field (MPDU/LPDU).

The transmitter terminates the packet by forcing the data output to be transitionless long enough for the receiver to recognize an invalid bit code. This signals the end of the packet. At the end of the packet transmission, the line must remain transitionless for three bit periods after the final clock transition.

The UART is responsible for providing the BitSync and ByteSync fields of the PPDU illustrated below. The layer two software manages all other encapsulating fields and provides these to the UART as part of the packet to be transmitted.

Bit Sync	Byte Sync	Priority	Alt Path	Delta BL	NPDU	CRC

Figure 49-338. Physical protocol data unit structure

NOTE

The maximum possible delay between the transmission of 3T line code violation from the end of final data bit and start of TB1T and TIDT timer is one sample clock.

49.4.1.2 Packet cycle and delay calculations

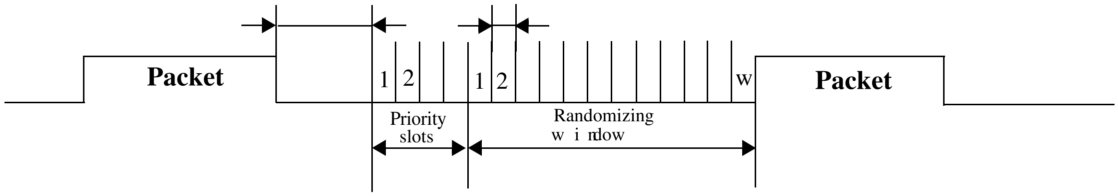


Figure 49-339. CEA709.1-B packet cycle

Predictive p-persistent CSMA is a technique for collision avoidance that randomizes channel access using knowledge of predicted load. It manages software using data and events reported by the hardware.

Beta1 delay is a value set by the software. It is generally a fixed value that is system dependent and hence does not usually require adjustment. It is measured in bit times, that is, the time that it takes for a single bit to be transmitted or one differential Manchester symbol. Beta1 is defined by CEA/EIA-709 specification as:

Beta1 > 1 bit time + $(2 \times T_{aup} + T_{aum})$

Where T_{aup} is the physical propagation delay defined by the media length.

T_{aum} is the detection and turn around-delay within the MAC sublayer; this is the period from the time the idle channel condition is detected, to the point when the first output transition appears on the output. On media where there is a carrier, this time must include the time between turning on the carrier, and it being asserted as a valid carrier on the medium.

The secondary delay timer is a value that is set by software. This is generally a variable value that must be set for each data message to be transmitted. It is measured in bit times, that is, the time that it takes for a single bit to be transmitted or one differential Manchester symbol. This value must be between 0 and (BL × Wbase) + (ProritySlots -1), Beta2 timeslots. A value of zero indicates that the queued packet for transmit is to be sent immediately upon expiration of the beta1 timer. According to the CEA/EIA-709 specification:

- BL is back log
- Wbase is 16 beta2 timeslots
- A priorityslot is the same amount of time as a beta2 timeslot
- Beta2 > $2 \times T_{aup} + T_{aum}$

Priority slots are handled completely by software. When calculating the secondary delay timer value, the software must take into account any priority slot that is included in the design of the system.

Each node must maintain an estimation of the current channel backlog. Backlog calculation is managed by the layer two software. Initially, the backlog is set to one. The backlog is incremented on transmission by a value indicated in the frames backlog increment field.

The backlog decrements under the following conditions:

- On waiting to transmit: If Wbase randomizing slots go by without channel activity.
- On receive: If a packet is received with a backlog increment of 0.
- On transmit: If a packet is transmitted with a backlog increment of 0.
- On idle: If a packet cycle time expires without channel activity.

The following actions need to be completed when a frame is received to prepare an outgoing message for transmission after the channel becomes idle:

- CRC of incoming message needs to be verified by software.
- If the CRC is good, the BL is recalculated, otherwise BL remains the same.
- Transmit delay (secondary delay timer) is calculated and supplied to UART.

49.4.1.3 Clock resynchronization

The UART is transmitting on time base source. Therefore, all receivers keep synchronizing with the node that is transmitting and no clock resynchronization occurs in transmitting.

When the UART is receiving or waiting to transmit, clock resynchronization is vital. Because long streams of data are possible (up to 229 bytes + headers), there exists significant potential for nodes to wander regarding time reference over the course of the message. Therefore, Differential Manchester Encoding (DME) is used. While DME requires twice the bandwidth of non-return to zero (NRZ) encoding schemes, it has the benefit of a guaranteed transition at the start of each bit transmitter. A transition occurring at the middle of the bit is encoded as a logic 0 or the lack of a transition at the middle of the bit time is encoded as a logic 1. By detecting the transition at the start of a bit period, the receiver is able to be resynthesized to the transmitter every bit period. Resynchronization can occur only after the node is already synchronized with the system. Additionally, for resynchronization to be effective, some basic assumptions regarding the system must be made:

1. Only a single channel sample may be in error (noise) over the entire bit (16 samples) period.
2. While a node is drifted from the system time base, with the resynchronization, the node is never shifted by more than 2 data samples in a given bit period.
3. If multiple noise events have occurred, no action is taken.

4. If a single noise event occurs, and it is possible to uniquely identify the noise event, then resynchronization takes place.

Starting at sample 15 of the previous time bit period, five data samples are collected. The number and location of the samples are key to decide if an adjustment in time base is required. Table below lists the possible values and the actions associated with each possibility. In the table, S means the data is the same as the logical value that was received in the second half of the previous bit period. D means that the sample is different from the logical value that was received in the second half of the previous bit period.

Sample Values (15,16,1,2,3)	Action / Event
SSSSS	No start of bit transition has been detected. Hence no adjustment to time base is made.
SSSSD	Two or more error events occurred or the time base was off. In this case, the time base is slowed down by two. Sample 3 becomes sample 1. The next sample is treated as sample 2.
SSSDS	Either two (or more) error events occurred, time base were off along with noise occurrence or sample 2 is noise and there is no start of bit transition. Hence no adjustment to time base is made.
SSSDD	It is possible that either noise was received during sample 1 or the time base needs shifting. In this case, the time base is slowed down by one. Sample 2 becomes sample 1, and sample 3 becomes sample 2. The next sample is treated as sample 3.
SSDSS	It is most likely that sample 1 is noise and there is no start of bit transition. Hence no adjustment to time base is made.
SSDSD	It is possible that sample 1 is noise, and time base needs shifting by two, or that sample 2 is noise. It is more likely that sample 2 is noise and therefore no adjustment to time base is made.
SSDDS	It is most likely that sample 3 is noise. Therefore, no adjustment to time base is made.
SSDDD	This is the expected case. Therefore, no adjustment to time base is made.
SDSSS	It is most likely that sample 16 is noise and there is no start of bit transition, hence no adjustment to time base is made.
SDSSD	Either multiple errors occurred or sample 16 is noise and time base is off by two. In this case, the time base is slowed down by two. Sample 3 becomes sample 1. The next sample is treated as sample 2.
SDSDS	In this case multiple errors have occurred. Hence no adjustment to time base is made.
SDSDD	In this case, there must either be multiple noise or one noise (at sample 16 or sample 1) with a time shift. Assuming that one noise occurred, it unclear what direction the time shift is. Hence no adjustment to time base is made.

Table continues on the next page...

Sample Values (15,16,1,2,3)	Action / Event
SDDSS	In this case either multiple errors occurred, two (or more) noise or two (or more) noise and a time shift. The most likely case is that samples 16 and 1 are noise. Hence no adjustment to time base is made.
SDDSD	The most likely case is noise for sample 2 and a time shift. Therefore, the time base is sped up by one. Sample 16 becomes sample 1, sample 1 becomes sample 2, sample 2 becomes sample 3, sample 3 becomes sample 4, and the next sample taken is sample 5.
SDDDS	The most likely case is noise for sample 3 and a time shift. Therefore, the time base is sped up by one. Sample 16 becomes sample 1, sample 1 becomes sample 2, sample 2 becomes sample 3, sample 3 becomes sample 4, and the next sample taken is sample 5.
SDDDD	Either sample 16 is noise or the time base has shifted. In this case, it is assumed that a time shift has occurred. Therefore, the time base is sped up by one. Sample 15 becomes sample 1, sample 1 becomes sample 2, sample 2 becomes sample 3, sample 3 becomes sample 4, and the next sample taken is sample 5.
DSSSS	It is most likely that sample 16 is noise and there is no start of bit transition, hence no adjustment to time base is made.
DSSSD	It is most likely that sample 15 is noise along with time shift. In this case, the time base is slowed down by two. Sample 3 becomes sample 1. The next sample is treated as sample 2.
DSSDS	In this case multiple errors occurred. Hence no adjustment to time base is made.
DSSDD	Either multiple errors occurred, possibly with time shift, or sample 15 is noise. In this case, the time base is slowed down by one. Sample 2 becomes sample 1, and sample 3 becomes sample 2. The next sample is treated as sample 3.
DSDSS	In this case multiple errors occurred. Hence no adjustment to time base is made.
DSDSD	In this case, multiple errors occurred. Therefore, no adjustment to time base is made.
DSDDS	In this case, multiple errors occurred. Therefore, no adjustment to time base is made.
DSDDD	In this case, either multiple errors occurred or sample 15 is noise and there is no start of bit transition. Therefore, no adjustment to time base is made.
DDSSS	In this case multiple errors occurred. It is most likely that samples 15 and 16 are noise. Hence no adjustment to time base is made.
DDSSD	In this case multiple errors occurred. Hence no adjustment to time base is made.
DDSDS	In this case multiple errors occurred. Hence no adjustment to time base is made.

Table continues on the next page...

Sample Values (15,16,1,2,3)	Action / Event
DDSDD	It is most likely that sample 1 is noise. Therefore, the time base is sped up by two. Sample 15 becomes sample 1, sample 16 becomes sample 2, sample 1 becomes sample 3, sample 2 becomes sample 4, sample 3 becomes sample 5, and the next sample taken is sample 6.
DDDSS	In this case multiple errors occurred along with time shift. Hence no adjustment to time base is made.
DDDSD	It is most likely that sample 2 is noise. Therefore, the time base is sped up by two. Sample 15 becomes sample 1, sample 16 becomes sample 2, sample 1 becomes sample 3, sample 2 becomes sample 4, sample 3 becomes sample 5, and the next sample taken is sample 6.
DDDDS	It is most likely that sample 3 is noise. Therefore, the time base is sped up by two. Sample 15 becomes sample 1, sample 16 becomes sample 2, sample 1 becomes sample 3, sample 2 becomes sample 4, sample 3 becomes sample 5, and the next sample taken is sample 6.
DDDDD	Either samples 15 and 16 are noise or the time base has shifted. Therefore, the time base is speed up by two. Sample 15 becomes sample 1, sample 16 becomes sample 2, sample 1 becomes sample 3, sample 2 becomes sample 4, sample 3 becomes sample 5, and the next sample taken is sample 6.

49.4.1.4 Data sampling

The receiver samples the unsynchronized receiver input signal at the RT clock rate. The RT clock is an internal signal with a frequency 16 times the baud rate. To adjust for baud rate mismatch, the RT clock is resynchronized after every bit.

To locate the start of preamble, data recovery logic does an asynchronous search for a valid edge, logic 0 preceded by two logic 1s or logic 1 preceded by two logic 0s. When the falling edge or rising edge of a possible preamble bit occurs, the RT clock begins to count to 16.

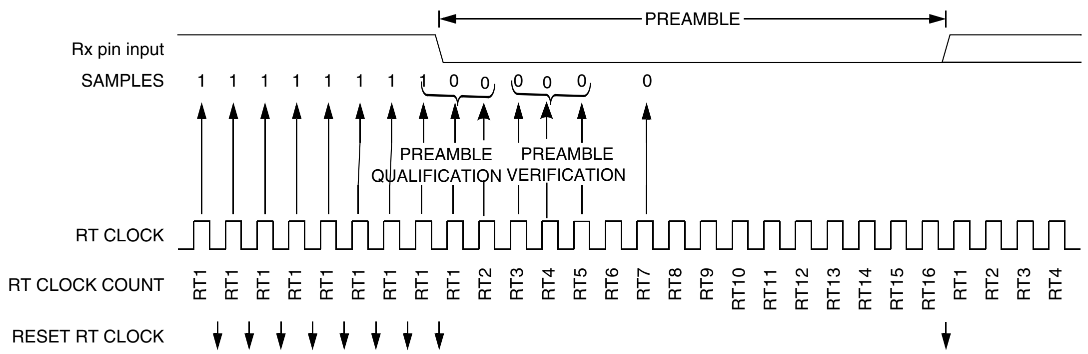


Figure 49-340. Receiver data sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT4, and RT5. The following table summarizes the results of the preamble verification samples.

Table 49-347. Preamble/ Data bit verification

RT3, RT4, and RT5 samples	Preamble verification
000	Yes
001	Yes
010	Yes
011	No
100	Yes
101	No
110	No
111	No

If preamble verification is not successful, the RT clock is reset and a new search for a preamble begins.

To determine the value of a data bit, recovery logic takes samples at RT11, RT12, and RT13. The following table summarizes the results of the data bit samples. If the majority of RT11, RT12, and RT13 samples is the same as the majority of RT3, RT4, and RT5 samples, then the data bit detected is 1, else the data bit detected is 0.

Table 49-348. Data bit recovery

RT11, RT12, and RT13 samples	Data bit determination
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

To signify the end of a data packet, the transmitter causes a line-code violation to occur, that is, the transmitter remains transitionless for at least 3-bit periods after the final clock transition, excluding the final data transition, if it exists. The receiver detects this violation. For the purpose of detecting a line-code violation, the receiver monitors the channel to locate a series of five or six back-to-back half bit periods.

49.4.1.5 Initial clock synchronization

When operating with EN709 set, there are various times when initial clock synchronization is required. When the UART has just been enabled, there is clearly no system clock reference. Additionally, if a channel has remained idle for a significant period of time, such as the arbitration time between packets, substantial clock drift may have occurred in the system between nodes. This is because there have been meaningful clock transitions on the channel to keep nodes synchronized. After these events, the clock may require significant synchronization adjustment; this event is referred to as initial clock synchronization.

There are three situations that may occur when a node attempts to obtain initial clock synchronization.

1. The node enters the system while a data packet is being actively transmitted.
2. The node enters the system while there is no data packet being actively transmitted on the system.
3. The node is already in the system and initial clock synchronization is required due to the end of a packet.

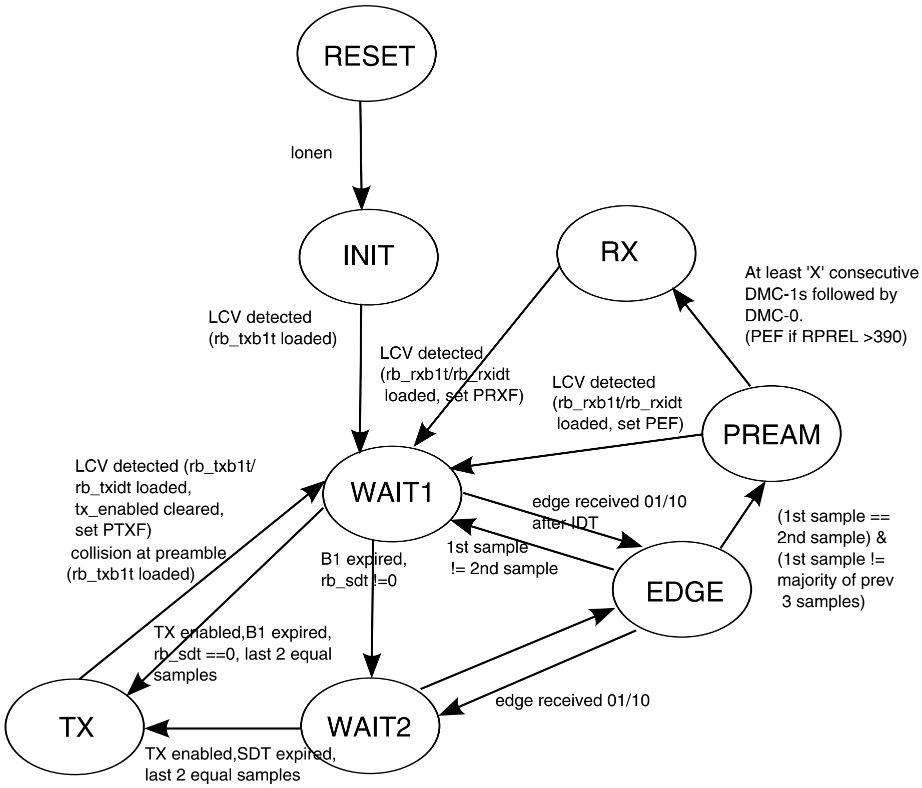
For case 1 and 2, the UART implements the following procedure:

1. The UART attempts to identify a valid line code violation. The line code violation is detected if the LCV counter reached the programmed value (default 20). The LCV counter resets automatically whenever valid edge is identified. Valid edge is identified if there is a transition in samples i.e. 0->1(sample1) or 1->0(sample1), and the subsequent two samples are equal (i.e sample 1 and sample 2) and sample 1 is not equal to the majority of previous 3 samples with respect to sample 1.
2. If the required line-code violation is detected, the beta1 delay timer will start and the UART will transit to case 3. Until then the UART will not be able to receive nor transmit any packet.

For case 3, it implements the following procedure:

1. Beta1 delay and secondary delay times increment as appropriate, that is Beta1 delay expires before the secondary delay timer starts.
2. While the timers are counting, the UART attempts to identify a valid edge.
3. If a valid receive edge is identified after the appropriate IDT has expired (RIDT or TIDT), but before the SDT expires and data is queued to be transmitted (C6[TX709] set), the transmission will be aborted before starting and the incoming data packet will be received. The S3[TXFF] flag will set to indicate that the transmission did not start.

4. If a valid edge is not identified before the delay time expires, and data is queued to be transmitted, the UART considers itself synchronized, and starts the preamble process.
5. If a valid edge is not identified before the delay time expires, and data is not queued to be transmitted, the UART continues attempting to locate a valid edge using the same process, and receives the incoming data packet like in step 3.



49.4.1.6 Priority packet preemption

The first data is fetched from the data buffer immediately after the preamble has completed. Therefore, it is possible to decide which data is sent during transmission until the completion of the preamble. This can be done in two different ways.

- The expected data to be transmitted can be written to the data buffer before or shortly after TE is enabled. In this case, the data is ready before the start of the preamble period. If a high priority packet has been identified for immediate/preemptive transmission, software may flush the data buffer and put the new data into the data buffer. This new data must be put into the data buffer prior to the completion of the preamble. Similarly, the transmit packet length register needs to be updated.
- Software can trigger data to be transmitted by asserting TE before the actual data has been placed in the data buffer. In the end, the software can write data into the data

buffer and update the transmit packet length register. This occurs before the preamble completes. To assist in identifying how much time is left before the preamble completes, the preamble started interrupt is asserted when the UART starts transmitting the preamble.

NOTE

1. If the data buffer does not contain at least one byte of valid data and the transmit packet length register has been updated prior to the preamble completing, an underflow event will occur and TXEN is deasserted. The packet is terminated by transmitting line code violation.
2. The maximum possible delay between masking the RX (due to LON FSM entering into TX State) and the first transmit preamble edge is 1 bit time while OBE is asserted with no delay with respect to masking of RX line.

49.4.1.7 Collision detection

Collision flag is detected only when device is transmitting if C6[CE] is asserted. The collision pulse is valid if it is asserted for CPW number of peripheral clock cycles. If the collision signal is already asserted before the start of packet transmission, then the width of the collision pulse is calculated from the start of transmit packet as shown in the figure below. If the collision signal is not cleared by the software by writing 11b, then the flag continues to retain the previous value. After the flag is cleared, the collision pulse width is calculated again, and the flag is asserted, if the width is equal to or more than the programmed CPW value.

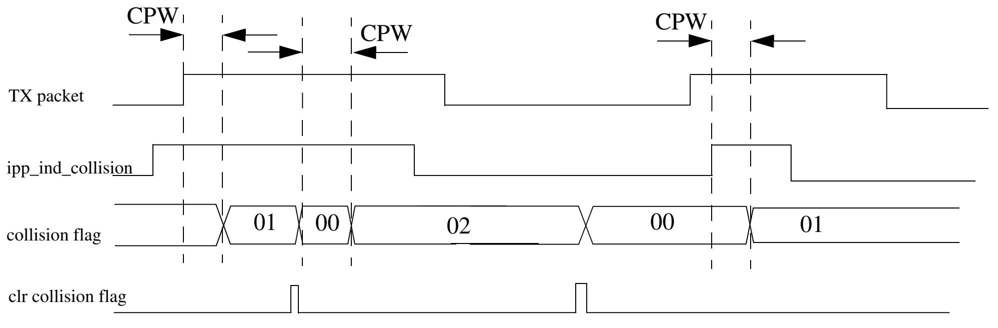


Figure 49-341. Collision pulse detection

The collision signal is asynchronous to the ipg clk, therefore the collision pulse of width exactly equal to CPW may not be detected correctly due to synchronization issue. The collision pulse visible to design may be decreased by one peripheral clock cycle due to the asynchronous nature of the collision pulse.

49.4.2 Transmitter

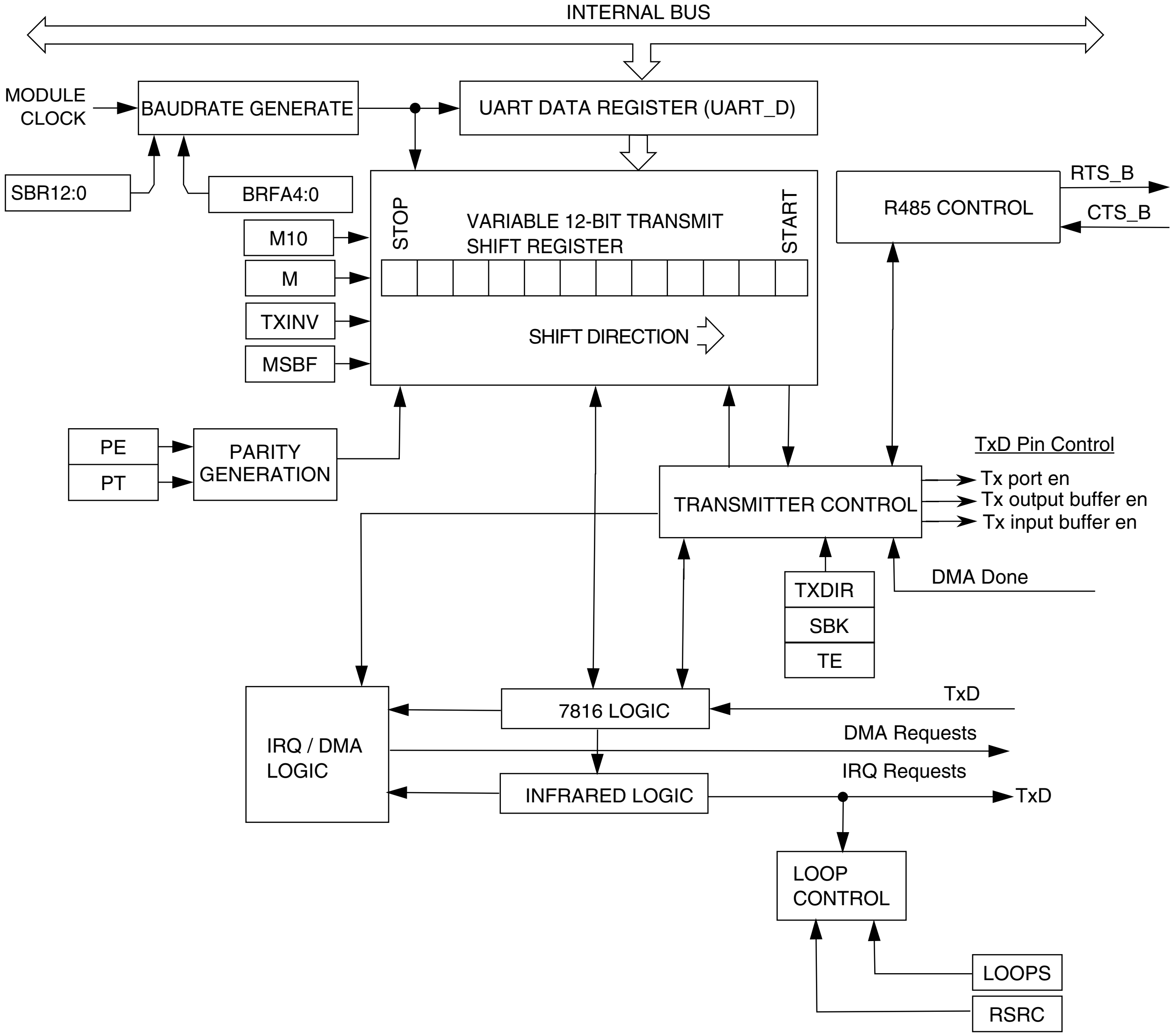


Figure 49-342. Transmitter Block Diagram

49.4.2.1 Transmitter character length

The UART transmitter can accommodate either 8, 9, or 10-bit data characters. The state of the C1[M] and C1[PE] bits and the C4[M10] bit determine the length of data characters. When transmitting 9-bit data, bit C3[T8] is the ninth bit (bit 8).

49.4.2.2 Transmission bit order

When S2[MSBF] is set, the UART automatically transmits the MSB of the data word as the first bit after the start bit. Similarly, the LSB of the data word is transmitted immediately preceding the parity bit, or the stop bit if parity is not enabled. All necessary bit ordering is handled automatically by the module. Therefore, the format of the data written to D for transmission is completely independent of the S2[MSBF] setting.

49.4.2.3 Character transmission

To transmit data, the MCU writes the data bits to the UART transmit buffer using UART data registers C3[T8] and D. Data in the transmit buffer is then transferred to the transmitter shift register as needed. The transmit shift register then shifts a frame out through the transmit data output signal after it has prefaced it with any required start and stop bits. The UART data registers, C3[T8] and D, provide access to the transmit buffer structure.

The UART also sets a flag, the transmit data register empty flag S1[TDRE], and generates an interrupt or DMA request (C5[TDMAS]) whenever the number of datawords in the transmit buffer is equal to or less than the value indicated by TWFIFO[TXWATER]. The transmit driver routine may respond to this flag by writing additional datawords to the transmit buffer using C3[T8]/D as space permits.

See Application information for specific programing sequences.

Setting C2[TE] automatically loads the transmit shift register with the following preamble:

- 10 logic 1s if C1[M] = 0
- 11 logic 1s if C1[M] = 1 and C4[M10] = 0
- 12 logic 1s if C1[M] = 1, C4[M10] = 1, C1[PE] = 1

After the preamble shifts out, control logic transfers the data from the D register into the transmit shift register. The transmitter automatically transmits the correct start bit and stop bit before and after the dataword.

When C7816[ISO_7816E] = 1, setting C2[TE] does not result in a preamble being generated. The transmitter starts transmitting as soon as the corresponding guard time expires. When C7816[TTYPE] = 0, the value in GT is used. When C7816[TTYPE] = 1, the value in BGT is used, because C2[TE] will remain asserted until the end of the block transfer. C2[TE] is automatically cleared when C7816[TTYPE] = 1 and the block being transmitted has completed. When C7816[TTYPE] = 0, the transmitter listens for a NACK indication. If no NACK is received, it is assumed that the character was correctly

received. If a NACK is received, the transmitter resends the data, assuming that the number of retries for that character, that is, the number of NACKs received, is less than or equal to the value in ET7816[TXTHRESHOLD].

Hardware supports odd or even parity. When parity is enabled, the bit immediately preceding the stop bit is the parity bit.

When the transmit shift register is not transmitting a frame, the transmit data output signal goes to the idle condition, logic 1. If at any time software clears C2[TE], the transmitter enable signal goes low and the transmit signal goes idle.

If the software clears C2[TE] while a transmission is in progress, the character in the transmit shift register continues to shift out, provided S1[TC] was cleared during the data write sequence. To clear S1[TC], the S1 register must be read followed by a write to D register.

If S1[TC] is cleared during character transmission and C2[TE] is cleared, the transmission enable signal is deasserted at the completion of the current frame. Following this, the transmit data out signal enters the idle state even if there is data pending in the UART transmit data buffer. To ensure that all the data written in the FIFO is transmitted on the link before clearing C2[TE], wait for S1[TC] to set. Alternatively, the same can be achieved by setting TWFIFO[TXWATER] to 0x0 and waiting for S1[TDRE] to set.

49.4.2.4 Transmitting break characters

Setting C2[SBK] loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on C1[M], C1[PE], S2[BRK13] and C4[M10]. See the following table.

Table 49-349. Transmit break character length

S2[BRK13]	C1[M]	C4[M10]	C1[PE]	Bits transmitted
0	0	—	—	10
0	1	1	0	11
0	1	1	1	12
1	0	—	—	13
1	1	—	—	14

As long as C2[SBK] is set, the transmitter logic continuously loads break characters into the transmit shift register. After the software clears C2[SBK], the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next character. Break bits are not supported when C7816[ISO_7816E] is set/enabled.

NOTE

When queuing a break character, it will be transmitted following the completion of the data value currently being shifted out from the shift register. This means that, if data is queued in the data buffer to be transmitted, the break character preempts that queued data. The queued data is then transmitted after the break character is complete.

49.4.2.5 Idle characters

An idle character contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on C1[M], C1[PE] and C4[M10]. The preamble is a synchronizing idle character that begins the first transmission initiated after setting C2[TE]. When C7816[ISO_7816E] is set/enabled, idle characters are not sent or detected. When data is not being transmitted, the data I/O line is in an inactive state.

If C2[TE] is cleared during a transmission, the transmit data output signal becomes idle after completion of the transmission in progress. Clearing and then setting C2[TE] during a transmission queues an idle character to be sent after the dataword currently being transmitted.

Note

When queuing an idle character, the idle character will be transmitted following the completion of the data value currently being shifted out from the shift register. This means that if data is queued in the data buffer to be transmitted, the idle character preempts that queued data. The queued data is then transmitted after the idle character is complete.

If C2[TE] is cleared and the transmission is completed, the UART is not the master of the TXD pin.

49.4.2.6 Hardware flow control

The transmitter supports hardware flow control by gating the transmission with the value of CTS. If the clear-to-send operation is enabled, the character is transmitted when CTS is asserted. If CTS is deasserted in the middle of a transmission with characters remaining in the receiver data buffer, the character in the shift register is sent and TXD remains in the mark state until CTS is reasserted.

If the clear-to-send operation is disabled, the transmitter ignores the state of CTS. Also, if the transmitter is forced to send a continuous low condition because it is sending a break character, the transmitter ignores the state of CTS regardless of whether the clear-to-send operation is enabled.

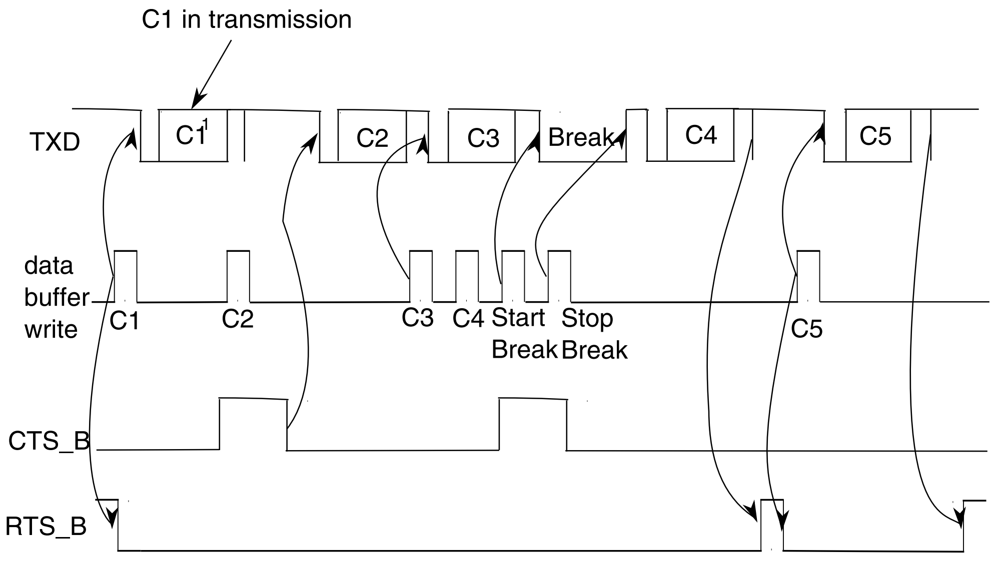
The transmitter's CTS signal can also be enabled even if the same UART receiver's RTS signal is disabled.

49.4.2.7 Transceiver driver enable

The transmitter can use RTS as an enable signal for the driver of an external transceiver. See Transceiver driver enable using RTS for details. If the request-to-send operation is enabled, when a character is placed into an empty transmitter data buffer, RTS asserts one bit time before the start bit is transmitted. RTS remains asserted for the whole time that the transmitter data buffer has any characters. RTS deasserts one bit time after all characters in the transmitter data buffer and shift register are completely sent, including the last stop bit. Transmitting a break character also asserts RTS, with the same assertion and deassertion timing as having a character in the transmitter data buffer.

The transmitter's RTS signal asserts only when the transmitter is enabled. However, the transmitter's RTS signal is unaffected by its CTS signal. RTS will remain asserted until the transfer is completed, even if the transmitter is disabled mid-way through a data transfer.

The following figure shows the functional timing information for the transmitter. Along with the actual character itself, TXD shows the start bit. The stop bit is also indicated, with a dashed line if necessary.



1. Cn = transmit characters

Figure 49-343. Transmitter RTS and CTS timing diagram

49.4.3 Receiver

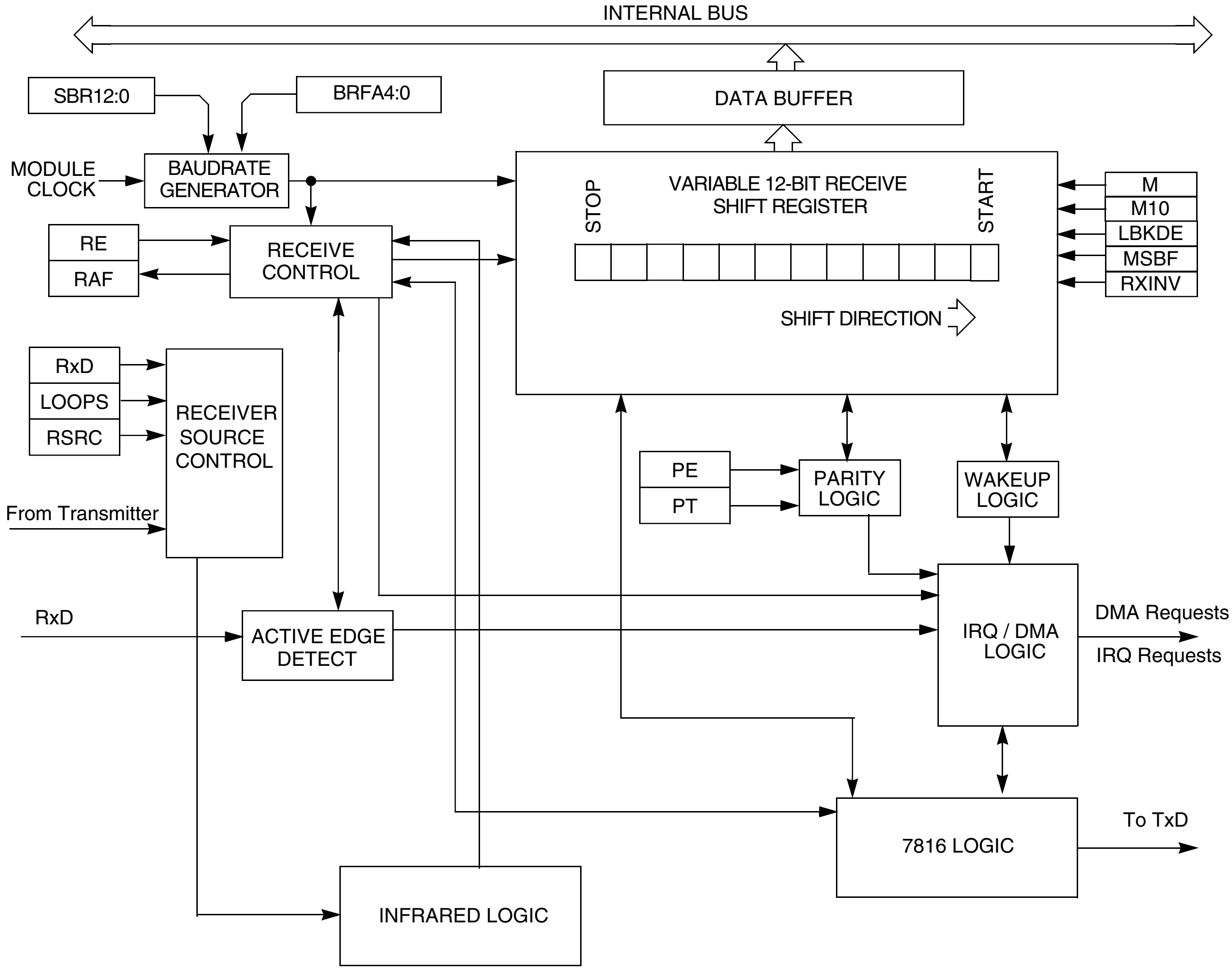


Figure 49-344. UART receiver block diagram

49.4.3.1 Receiver character length

The UART receiver can accommodate 8-, 9-, or 10-bit data characters. The states of C1[M], C1[PE] and C4[M10] determine the length of data characters. When receiving 9 or 10-bit data, C3[R8] is the ninth bit (bit 8).

49.4.3.2 Receiver bit ordering

When S2[MSBF] is set, the receiver operates such that the first bit received after the start bit is the MSB of the dataword. Similarly, the bit received immediately preceding the parity bit, or the stop bit if parity is not enabled, is treated as the LSB for the dataword. All necessary bit ordering is handled automatically by the module. Therefore, the format of the data read from receive data buffer is completely independent of S2[MSBF].

49.4.3.3 Character reception

During UART reception, the receive shift register shifts a frame in from the unsynchronized receiver input signal. After a complete frame shifts into the receive shift register, the data portion of the frame transfers to the UART receive buffer. Additionally, the noise and parity error flags that are calculated during the receive process are also captured in the UART receive buffer. The receive data buffer is accessible via the D and C3[T8] registers. Additional received information flags regarding the receive dataword can be read in ED register. S1[RDRF] is set if the number of resulting datawords in the receive buffer is equal to or greater than the number indicated by RWFIFO[RXWATER]. If the C2[RIE] is also set, RDRF generates an RDRF interrupt request. Alternatively, by programming C5[RDMAS], a DMA request can be generated.

When C7816[ISO_7816E] is set/enabled and C7816[TTYPE] = 0, character reception operates slightly differently. Upon receipt of the parity bit, the validity of the parity bit is checked. If C7816[ANACK] is set and the parity check fails, or if INIT and the received character is not a valid initial character, then a NACK is sent by the receiver. If the number of consecutive receive errors exceeds the threshold set by ET7816[RXTHRESHOLD], then IS7816[RXT] is set and an interrupt generated if IE7816[RXTE] is set. If an error is detected due to parity or an invalid initial character, the data is not transferred from the receive shift register to the receive buffer. Instead, the data is overwritten by the next incoming data.

When the C7816[ISO_7816E] is set/enabled, C7816[ONACK] is set/enabled, and the received character results in the receive buffer overflowing, a NACK is issued by the receiver. Additionally, S1[OR] is set and an interrupt is issued if required, and the data in the shift register is discarded.

49.4.3.4 Framing errors

If the data recovery logic does not detect a logic 1 where the stop bit should be in an incoming frame, it sets the framing error flag, S1[FE], if S2[LBKDE] is disabled. When S2[LBKDE] is disabled, a break character also sets the S1[FE] because a break character

has no stop bit. S1[FE] is set at the same time that received data is placed in the receive data buffer. Framing errors are not supported when C7816[ISO7816E] is set/enabled. However, if S1[FE] is set, data will not be received when C7816[ISO7816E] is set.

49.4.3.5 Receiving break characters

The UART recognizes a break character when a start bit is followed by eight, nine, or ten logic 0 data bits and a logic 0 where the stop bit should be. Receiving a break character has these effects on UART registers:

- Sets the framing error flag, S1[FE].
- Writes an all 0 dataword to the data buffer, which may cause S1[RDRF] to set, depending on the watermark and number of values in the data buffer.
- May set the overrun flag, S1[OR], noise flag, S1[NF], parity error flag, S1[PE], or the receiver active flag, S2[RAF].

The detection threshold for a break character can be adjusted when using an internal oscillator in a LIN system by setting S2[LBKDE]. The UART break character detection threshold depends on C1[M], C1[PE], S2[LBKDE] and C4[M10]. See the following table.

Table 49-350. Receive break character detection threshold

LBKDE	M	M10	PE	Threshold (bits)
0	0	—	—	10
0	1	0	—	11
0	1	1	1	12
1	0	—	—	11
1	1	—	—	12

While S2[LBKDE] is set, it will have these effects on the UART registers:

- Prevents S1[RDRF], S1[FE], S1[NF], and S1[PF] from being set. However, if they are already set, they will remain set.
- Sets the LIN break detect interrupt flag, S2[LBKDIF], if a LIN break character is received.

Break characters are not detected or supported when C7816[ISO_7816E] is set/enabled.

49.4.3.6 Hardware flow control

To support hardware flow control, the receiver can be programmed to automatically deassert and assert RTS.

- RTS remains asserted until the transfer is complete, even if the transmitter is disabled midway through a data transfer. See Transceiver driver enable using RTS for more details.
- If the receiver request-to-send functionality is enabled, the receiver automatically deasserts RTS if the number of characters in the receiver data register is equal to or greater than receiver data buffer's watermark, RWFIFO[RXWATER].
- The receiver asserts RTS when the number of characters in the receiver data register is less than the watermark. It is not affected if RDRF is asserted.
- Even if RTS is deasserted, the receiver continues to receive characters until the receiver data buffer is full or is overrun.
- If the receiver request-to-send functionality is disabled, the receiver RTS remains deasserted.

The following figure shows receiver hardware flow control functional timing. Along with the actual character itself, RXD shows the start bit. The stop bit can also indicated, with a dashed line, if necessary. The watermark is set to 2.

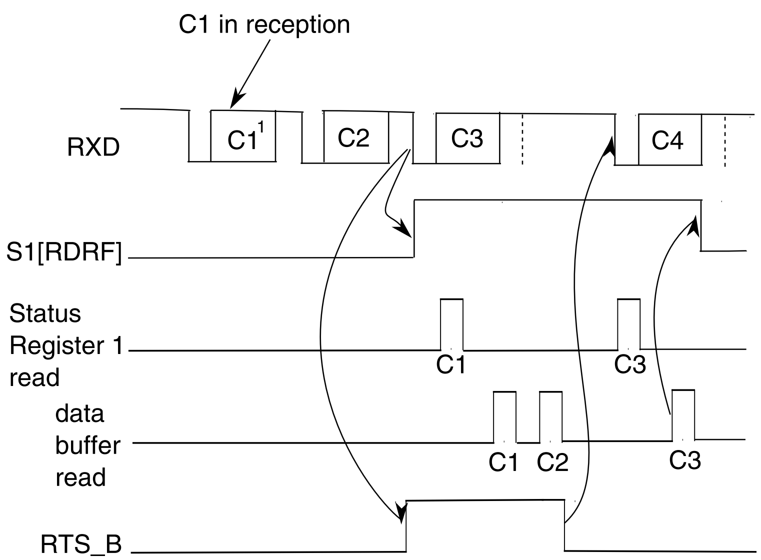


Figure 49-345. Receiver hardware flow control timing diagram

49.4.3.7 Infrared decoder

The infrared decoder converts the received character from the IrDA format to the NRZ format used by the receiver. It also has a 16-RT clock counter that filters noise and indicates when a 1 is received.

49.4.3.7.1 Start bit detection

When S2[RXINV] is cleared, the first rising edge of the received character corresponds to the start bit. The infrared decoder resets its counter. At this time, the receiver also begins its start bit detection process. After the start bit is detected, the receiver synchronizes its bit times to this start bit time. For the rest of the character reception, the infrared decoder's counter and the receiver's bit time counter count independently from each other.

49.4.3.7.2 Noise filtering

Any further rising edges detected during the first half of the infrared decoder counter are ignored by the decoder. Any pulses less than one RT clocks can be undetected by it regardless of whether it is seen in the first or second half of the count.

49.4.3.7.3 Low-bit detection

During the second half of the decoder count, a rising edge is decoded as a 0, which is sent to the receiver. The decoder counter is also reset.

49.4.3.7.4 High-bit detection

At 16-RT clocks after the previous rising edge, if a rising edge is not seen, then the decoder sends a 1 to the receiver.

If the next bit is a 0, which arrives late, then a low-bit is detected according to Low-bit detection. The value sent to the receiver is changed from 1 to a 0. Then, if a noise pulse occurs outside the receiver's bit time sampling period, then the delay of a 0 is not recorded as noise.

49.4.3.8 Baud rate tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples (RT8, RT9, and RT10) to fall outside the actual stop bit. A noise error will occur if the RT8, RT9, and RT10 samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the RT8, RT9, and RT10 stop bit samples are a logic 0.

As the receiver samples an incoming frame, it resynchronizes the RT clock on any valid falling edge within the frame. Resynchronization within frames corrects a misalignment between transmitter bit times and receiver bit times.

49.4.3.8.1 Slow data tolerance

The following figure shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

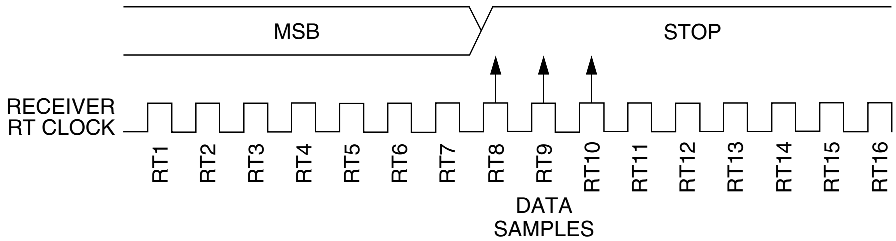


Figure 49-346. Slow data

For an 8-bit data character, data sampling of the stop bit takes the receiver 154 RT cycles (9 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 49-346, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 147 RT cycles (9 bit times × 16 RT cycles + 3 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

$$((154 - 147) \div 154) \times 100 = 4.54\%$$

For a 9-bit data character, data sampling of the stop bit takes the receiver 170 RT cycles (10 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 49-346, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 163 RT cycles (10 bit times × 16 RT cycles + 3 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

$$((170 - 163) \div 170) \times 100 = 4.12\%$$

49.4.3.8.2 Fast data tolerance

The following figure shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

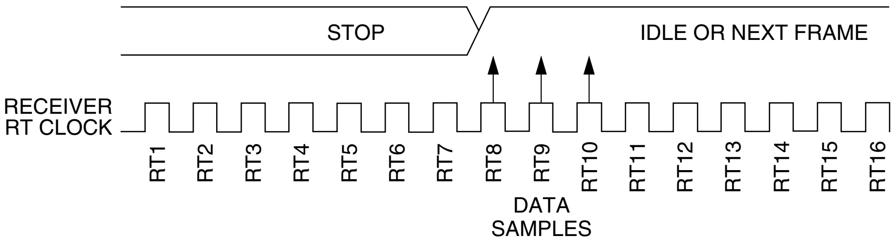


Figure 49-347. Fast data

For an 8-bit data character, data sampling of the stop bit takes the receiver 154 RT cycles (9 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 49-347, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 160 RT cycles (10 bit times × 16 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is:

$$((154 - 160) \div 154) \times 100 = 3.90\%$$

For a 9-bit data character, data sampling of the stop bit takes the receiver 170 RT cycles (10 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 49-347, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 176 RT cycles (11 bit times × 16 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

$$((170 - 176) \div 170) \times 100 = 3.53\%$$

49.4.3.9 Receiver wakeup

C1[WAKE] determines how the UART is brought out of the standby state to process an incoming message. C1[WAKE] enables either idle line wakeup or address mark wakeup.

Receiver wakeup is not supported when C7816[ISO_7816E] is set/enabled because multi-receiver systems are not allowed.

49.4.3.9.1 Idle input line wakeup (C1[WAKE] = 0)

In this wakeup method, an idle condition on the unsynchronized receiver input signal clears C2[RWU] and wakes the UART. The initial frame or frames of every message contain addressing information. All receivers evaluate the addressing information, and receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its C2[RWU] and return to the standby state. C2[RWU] remains set and the receiver remains in standby until another idle character appears on the unsynchronized receiver input signal.

Idle line wakeup requires that messages be separated by at least one idle character and that no message contains idle characters.

When C2[RWU] is 1 and S2[RWUID] is 0, the idle character that wakes the receiver does not set S1[IDLE] or the receive data register full flag, S1[RDRF]. The receiver wakes and waits for the first data character of the next message which is stored in the receive data buffer. When S2[RWUID] and C2[RWU] are set and C1[WAKE] is cleared, any idle condition sets S1[IDLE] and generates an interrupt if enabled.

Idle input line wakeup is not supported when C7816[ISO_7816E] is set/enabled.

49.4.3.9.2 Address mark wakeup (C1[WAKE] = 1)

In this wakeup method, a logic 1 in the bit position immediately preceding the stop bit of a frame clears C2[RWU] and wakes the UART. A logic 1 in the bit position immediately preceeding the stop bit marks a frame as an address frame that contains addressing information. All receivers evaluate the addressing information, and the receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its C2[RWU] and return to the standby state. C2[RWU] remains set and the receiver remains in standby until another address frame appears on the unsynchronized receiver input signal.

A logic 1 in the bit position immediately preceding the stop bit clears the receiver's C2[RWU] after the stop bit is received and places the received data into the receiver data buffer. Note that if Match Address operation is enabled i.e. C4[MAEN1] or C4[MAEN2] is set, then received frame is transferred to receive buffer only if the comparison matches.

Address mark wakeup allows messages to contain idle characters but requires that the bit position immediately preceding the stop bit be reserved for use in address frames.

If module is in standby mode and nothing triggers to wake the UART, no error flag is set even if an invalid error condition is detected on the receiving data line.

Address mark wakeup is not supported when C7816[ISO_7816E] is set/enabled.

49.4.3.9.3 Match address operation

Match address operation is enabled when C4[MAEN1] or C4[MAEN2] is set. In this function, a frame received by the RX pin with a logic 1 in the bit position of the address mark is considered an address and is compared with the associated MA1 or MA2 register. The frame is transferred to the receive buffer, and S1[RDRF] is set, only if the comparison matches. All subsequent frames received with a logic 0 in the bit position of the address mark are considered to be data associated with the address and are transferred to the receive data buffer. If no marked address match occurs, then no transfer is made to the receive data buffer, and all following frames with logic 0 in the bit position of the address mark are also discarded. If both C4[MAEN1] and C4[MAEN2] are negated, the receiver operates normally and all data received is transferred to the receive data buffer.

Match address operation functions in the same way for both MA1 and MA2 registers. Note that the position of the address mark is the same as the Parity Bit when parity is enabled for 8 bit and 9 bit data formats.

- If only one of C4[MAEN1] and C4[MAEN2] is asserted, a marked address is compared only with the associated match register and data is transferred to the receive data buffer only on a match.
- If C4[MAEN1] and C4[MAEN2] are asserted, a marked address is compared with both match registers and data is transferred only on a match with either register.

Address match operation is not supported when C7816[ISO_7816E] is set/enabled.

49.4.4 Baud rate generation

A 13-bit modulus counter and a 5-bit fractional fine-adjust counter in the baud rate generator derive the baud rate for both the receiver and the transmitter. The value from 1 to 8191 written to SBR[12:0] determines the module clock divisor. The SBR bits are in the UART baud rate registers, BDH and BDL. The baud rate clock is synchronized with the module clock and drives the receiver. The fractional fine-adjust counter adds fractional delays to the baud rate clock to allow fine trimming of the baud rate to match the system baud rate. The transmitter is driven by the baud rate clock divided by 16. The receiver has an acquisition rate of 16 samples per bit time.

Baud rate generation is subject to two sources of error:

- Integer division of the module clock may not give the exact target frequency. This error can be reduced with the fine-adjust counter.
- Synchronization with the module clock can cause phase shift.

The Table 49-351 lists the available baud divisor fine adjust values.

UART baud rate = UART module clock / (16 × (SBR[12:0] + BRFD))

The following table lists some examples of achieving target baud rates with a module clock frequency of 10.2 MHz, with and without fractional fine adjustment.

Table 49-351. Baud rates (example: module clock = 10.2 MHz)

Bits SBR (decimal)	Bits BRFA	BRFD value	Receiver clock (Hz)	Transmitter clock (Hz)	Target Baud rate	Error (%)
17	00000	0	600,000.0	37,500.0	38,400	2.3
16	10011	19/32=0.59375	614,689.3	38,418.08	38,400	0.047
33	00000	0	309,090.9	19,318.2	19,200	0.62
33	00110	6/32=0.1875	307,344.6	19,209.04	19,200	0.047
66	00000	0	154,545.5	9659.1	9600	0.62
133	00000	0	76,691.7	4793.2	4800	0.14
266	00000	0	38,345.9	2396.6	2400	0.14
531	00000	0	19,209.0	1200.6	1200	0.11
1062	00000	0	9604.5	600.3	600	0.05
2125	00000	0	4800.0	300.0	300	0.00
4250	00000	0	2400.0	150.0	150	0.00
5795	00000	0	1760.1	110.0	110	0.00

Table 49-352. Baud rate fine adjust

BRFA	Baud Rate Fractional Divisor (BRFD)
0 0 0 0 0	0/32 = 0
0 0 0 0 1	1/32 = 0.03125
0 0 0 1 0	2/32 = 0.0625
0 0 0 1 1	3/32 = 0.09375
0 0 1 0 0	4/32 = 0.125
0 0 1 0 1	5/32 = 0.15625
0 0 1 1 0	6/32 = 0.1875
0 0 1 1 1	7/32 = 0.21875
0 1 0 0 0	8/32 = 0.25
0 1 0 0 1	9/32 = 0.28125
0 1 0 1 0	10/32 = 0.3125
0 1 0 1 1	11/32 = 0.34375
0 1 1 0 0	12/32 = 0.375
0 1 1 0 1	13/32 = 0.40625
0 1 1 1 0	14/32 = 0.4375

Table continues on the next page...

Table 49-352. Baud rate fine adjust (continued)

BRFA	Baud Rate Fractional Divisor (BRFD)
0 1 1 1 1	15/32 = 0.46875
1 0 0 0 0	16/32 = 0.5
1 0 0 0 1	17/32 = 0.53125
1 0 0 1 0	18/32 = 0.5625
1 0 0 1 1	19/32 = 0.59375
1 0 1 0 0	20/32 = 0.625
1 0 1 0 1	21/32 = 0.65625
1 0 1 1 0	22/32 = 0.6875
1 0 1 1 1	23/32 = 0.71875
1 1 0 0 0	24/32 = 0.75
1 1 0 0 1	25/32 = 0.78125
1 1 0 1 0	26/32 = 0.8125
1 1 0 1 1	27/32 = 0.84375
1 1 1 0 0	28/32 = 0.875
1 1 1 0 1	29/32 = 0.90625
1 1 1 1 0	30/32 = 0.9375
1 1 1 1 1	31/32 = 0.96875

49.4.5 Data format (non ISO-7816)

Each data character is contained in a frame that includes a start bit and a stop bit. The rest of the data format depends upon C1[M], C1[PE], S2[MSBF] and C4[M10].

49.4.5.1 Eight-bit configuration

Clearing C1[M] configures the UART for 8-bit data characters, that is, eight bits are memory mapped in D. A frame with eight data bits has a total of 10 bits. The most significant bit of the eight data bits can be used as an address mark to wake the receiver. If the most significant bit is used in this way, then it serves as an address or data indication, leaving the remaining seven bits as actual data. When C1[PE] is set, the eighth data bit is automatically calculated as the parity bit. See the following table.

Table 49-353. Configuration of 8-bit data format

UART_C1[PE]	Start bit	Data bits	Address bits	Parity bits	Stop bit
0	1	8	0	0	1

Table continues on the next page...

Table 49-353. Configuration of 8-bit data format (continued)

UART_C1[PE]	Start bit	Data bits	Address bits	Parity bits	Stop bit
0	1	7	1[1]	0	1
1	1	7	0	1	1

1. The address bit identifies the frame as an address character. See Receiver wakeup.

49.4.5.2 Nine-bit configuration

When C1[M] is set and C4[M10] is cleared, the UART is configured for 9-bit data characters. If C1[PE] is enabled, the ninth bit is either C3[T8/R8] or the internally generated parity bit. This results in a frame consisting of a total of 11 bits. In the event that the ninth data bit is selected to be C3[T8], it will remain unchanged after transmission and can be used repeatedly without rewriting it, unless the value needs to be changed. This feature may be useful when the ninth data bit is being used as an address mark.

When C1[M] and C4[M10] are set, the UART is configured for 9-bit data characters, but the frame consists of a total of 12 bits. The 12 bits include the start and stop bits, the 9 data character bits, and a tenth internal data bit. Note that if C4[M10] is set, C1[PE] must also be set. In this case, the tenth bit is the internally generated parity bit. The ninth bit can either be used as an address mark or a ninth data bit.

See the following table.

Table 49-354. Configuration of 9-bit data formats

C1[PE]	UC1[M]	C1[M10]	Start bit	Data bits	Address bits	Parity bits	Stop bit
0	0	0	See Eight-bit configuration				
0	0	1	Invalid configuration				
0	1	0	1	9	0	0	1
0	1	0	1	8	1[1]	0	1
0	1	1	Invalid Configuration				
1	0	0	See Eight-bit configuration				
1	0	1	Invalid Configuration				
1	1	0	1	8	0	1	1
1	1	1	1	9	0	1	1
1	1	1	1	8	1[2]	1	1

1. The address bit identifies the frame as an address character.
2. The address bit identifies the frame as an address character.

Note

Unless in 9-bit mode with M10 set, do not use address mark wakeup with parity enabled.

49.4.5.3 Timing examples

Timing examples of these configurations in the NRZ mark/space data format are illustrated in the following figures. The timing examples show all of the configurations in the following sub-sections along with the LSB and MSB first variations.

49.4.5.3.1 Eight-bit format with parity disabled

The most significant bit can be used for address mark wakeup.

Figure 49-348. Eight bits of data with LSB first

Figure 49-349. Eight bits of data with MSB first

49.4.5.3.2 Eight-bit format with parity enabled

Figure 49-350. Seven bits of data with LSB first and parity

Figure 49-351. Seven bits of data with MSB first and parity

49.4.5.3.3 Nine-bit format with parity disabled

The most significant bit can be used for address mark wakeup.

Figure 49-352. Nine bits of data with LSB first

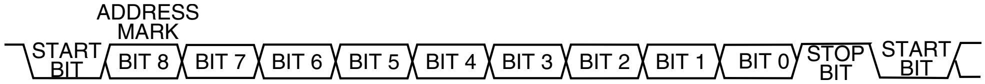


Figure 49-353. Nine bits of data with MSB first

49.4.5.3.4 Nine-bit format with parity enabled

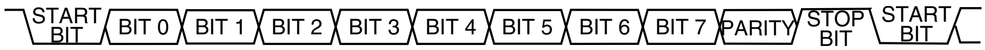


Figure 49-354. Eight bits of data with LSB first and parity

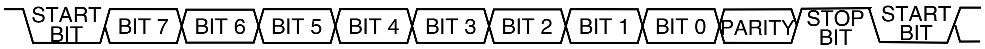


Figure 49-355. Eight bits of data with MSB first and parity

49.4.5.3.5 Non-memory mapped tenth bit for parity

The most significant memory-mapped bit can be used for address mark wakeup.

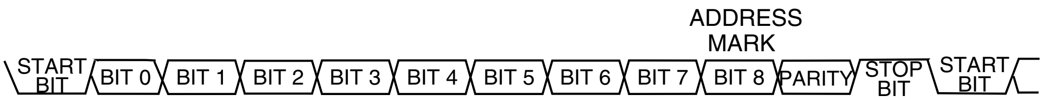


Figure 49-356. Nine bits of data with LSB first and parity

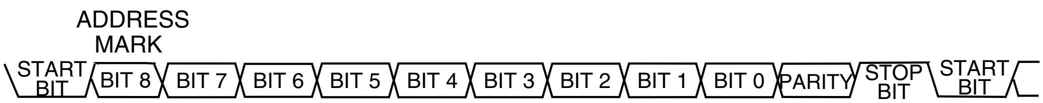


Figure 49-357. Nine bits of data with MSB first and parity

49.4.6 Single-wire operation

Normally, the UART uses two pins for transmitting and receiving. In single wire operation, the RXD pin is disconnected from the UART and the UART implements a half-duplex serial connection. The UART uses the TXD pin for both receiving and transmitting.

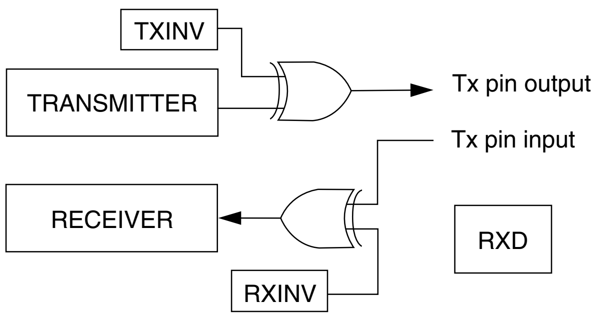


Figure 49-358. Single-wire operation (C1[LOOPS] = 1, C1[RSRC] = 1)

Enable single wire operation by setting C1[LOOPS] and the receiver source field, C1[RSRC]. Setting C1[LOOPS] disables the path from the unsynchronized receiver input signal to the receiver. Setting C1[RSRC] connects the receiver input to the output of the TXD pin driver. Both the transmitter and receiver must be enabled (C2[TE] = 1 and C2[RE] = 1). When C7816[ISO_7816EN] is set, it is not required that both C2[TE] and C2[RE] are set.

49.4.7 Loop operation

In loop operation, the transmitter output goes to the receiver input. The unsynchronized receiver input signal is disconnected from the UART.

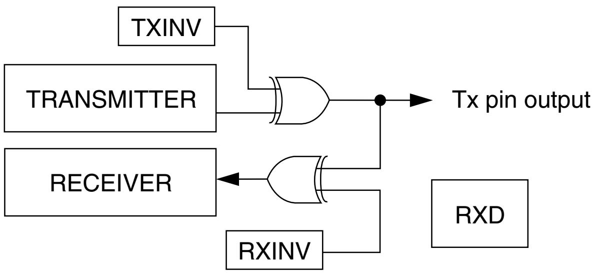


Figure 49-359. Loop operation (C1[LOOPS] = 1, C1[RSRC] = 0)

Enable loop operation by setting C1[LOOPS] and clearing C1[RSRC]. Setting C1[LOOPS] disables the path from the unsynchronized receiver input signal to the receiver. Clearing C1[RSRC] connects the transmitter output to the receiver input. Both the transmitter and receiver must be enabled (C2[TE] = 1 and C2[RE] = 1). When C7816[ISO_7816EN] is set, it is not required that both C2[TE] and C2[RE] are set.

49.4.8 ISO-7816/smartcard support

The UART provides mechanisms to support the ISO-7816 protocol that is commonly used to interface with smartcards. The ISO-7816 protocol is an NRZ, single wire, half-duplex interface. The TxD pin is used in open-drain mode because the data signal is used for both transmitting and receiving. There are multiple subprotocols within the ISO-7816 standard. The UART supports both T = 0 and T = 1 protocols. The module also provides for automated initial character detection and configuration, which allows for support of both direct convention and inverse convention data formats. A variety of interrupts specific to 7816 are provided in addition to the general interrupts to assist software. Additionally, the module is able to provide automated NACK responses and has programmed automated retransmission of failed packets. An assortment of programmable timeouts and guard band times are also supported.

The term elemental time unit (ETU) is frequently used in the context of ISO-7816. This concept is used to relate the frequency that the system (UART) is running at and the frequency that data is being transmitted and received. One ETU represents the time it takes to transmit or receive a single bit. For example, a standard 7816 packet, excluding any guard time or NACK elements is 10 ETUs (start bit, 8 data bits, and a parity bit). Guard times and wait times are also measured in ETUs.,

NOTE

The ISO-7816 specification may have certain configuration options that are reserved. To maintain maximum flexibility to support future 7816 enhancements or devices that may not strictly conform to the specification, the UART does not prevent those options being used. Further, the UART may provide configuration options that exceed the flexibility of options explicitly allowed by the 7816 specification. Failure to correctly configure the UART may result in unexpected behavior or incompatibility with the ISO-7816 specification.

49.4.8.1 Initial characters

In ISO-7816 with T = 0 mode, the UART can be configured to use C7816[INIT] to detect the next valid initial character, referred to by the ISO-7816 specifically as a TS character. When the initial character is detected, the UART provides the host processor with an interrupt if IE7816[INITDE] is set. Additionally, the UART will alter S2[MSBF], C3[TXINV], and S2[RXINV] automatically, based on the initial character. The corresponding initial character and resulting register settings are listed in the following table.

Table 49-355. Initial character automated settings

Initial character (bit 1-10)	Initial character (hex)	MSBF	TXINV	RXINV
LHHL LLL LLH inverse convention	3F	1	1	1
LHHL HHH LLH direct convention	3B	0	0	0

S2[MSBF], C3[TXINV], and S2[RXINV] must be reset to their default values before C7816[INIT] is set. Once C7816[INIT] is set, the receiver searches all received data for the first valid initial character. Detecting a Direct Convention Initial Character will cause no change to S2[MSBF], C3[TXINV], and S2[RXINV], while detecting an Inverse Convention Initial Character will cause these fields to set automatically. All data

received, which is not a valid initial character, is ignored and all flags resulting from the invalid data are blocked from asserting. If C7816[ANACK] is set, a NACK is returned for invalid received initial characters and an RXT interrupt is generated as programmed.

49.4.8.2 Protocol T = 0

When T = 0 protocol is selected, a relatively complex error detection scheme is used. Data characters are formatted as illustrated in the following figure. This scheme is also used for answer to reset and Peripheral Pin Select (PPS) formats.

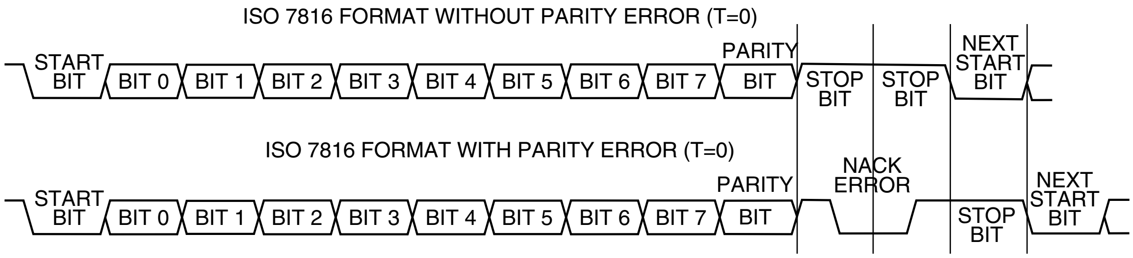


Figure 49-360. ISO-7816 T = 0 data format

As with other protocols supported by the UART, the data character includes a start bit. However, in this case, there are two stop bits rather than the typical single stop bit. In addition to a standard even parity check, the receiver has the ability to generate and return a NACK during the second half of the first stop bit period. The NACK must be at least one time period (ETU) in length and no more than two time periods (ETU) in length. The transmitter must wait for at least two time units (ETU) after detection of the error signal before attempting to retransmit the character.

It is assumed that the UART and the device (smartcard) know in advance which device is receiving and which is transmitting. No special mechanism is supplied by the UART to control receive and transmit in the mode other than C2[TE] and C2[RE]. Initial Character Detect feature is also supported in this mode.

49.4.8.3 Protocol T = 1

When T = 1 protocol is selected, the NACK error detection scheme is not used. Rather, the parity bit is used on a character basis and a CRC or LRC is used on the block basis, that is, for each group of characters. In this mode, the data format allows for a single stop bit although additional inactive bit periods may be present between the stop bit and the next start bit. Data characters are formatted as illustrated in the following figure.

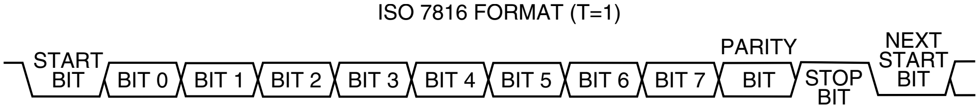


Figure 49-361. ISO 7816 T=1 data format

The smallest data unit that is transferred is a block. A block is made up of several data characters and may vary in size depending on the block type. The UART does not provide a mechanism to decode the block type. As part of the block, an LRC or CRC is included. The UART does not calculate the CRC or LRC for transmitted blocks, nor does it verify the validity of the CRC or LRC for received blocks. The 7816 protocol requires that the initiator and the smartcard (device) takes alternate turns in transmitting and receiving blocks. When the UART detects that the last character in a block has been transmitted it will automatically clear C2[TE], C3[TXDIR] and enter receive mode. Therefore, the software must program the transmit buffer with the next data to be transmitted and then enable C2[TE] and set C3[TXDIR], once the software has determined that the last character of the received block has been received. The UART detects that the last character of the transmit block has been sent when TL7816[TLEN] = 0 and four additional characters have been sent. The four additional characters are made up of three prior to TL7816[TLEN] decrementing (prologue) and one after TL7816[TLEN] = 0, the final character of the epilogue.

49.4.8.4 Wait time and guard time parameters

The ISO-7816 specification defines several wait time and guard time parameters. The UART allows for flexible configuration and violation detection of these settings. On reset, the wait time (IS7816[WT]) defaults to 9600 ETUs and guard time (GT) to 12 ETUs. These values are controlled by parameters in the WP7816, WN7816, and WF7816 registers. Additionally, the value of C7816[TTYPE] also factors into the calculation. The formulae used to calculate the number ETUs for each wait time and guard time value are shown in Table 49-356.

Wait time (WT) is defined as the maximum allowable time between the leading edge of a character transmitted by the smartcard device and the leading edge of the previous character that was transmitted by the UART or the device. Similarly, character wait time (CWT) is defined as the maximum allowable time between the leading edge of two characters within the same block. Block wait time (BWT) is defined as the maximum time between the leading edge character of the last block received by the smartcard device and the leading edge of the first character transmitted by the smartcard device.

Guard time (GT) is defined as the minimum allowable time between the leading edge of two consecutive characters. Character guard time (CGT) is the minimum allowable time between the leading edges of two consecutive characters in the same direction, that is,

transmission or reception. Block guard time (BGT) is the minimum allowable time between the leading edges of two consecutive characters in opposite directions, that is, transmission then reception or reception then transmission.

The GT and WT counters reset whenever C7816[TTYPE] = 1 or C7816[ISO_7816E] = 0 or a new dataword start bit has been received or transmitted as specified by the counter descriptions. The CWT, CGT, BWT, BGT counters reset whenever C7816[TTYPE] = 0 or C7816[ISO_7816E] = 0 or a new dataword start bit is received or transmitted as specified by the counter descriptions. When C7816[TTYPE] = 1, some of the counter values require an assumption regarding the first data transferred when the UART first starts. This assumption is required when the 7816E is disabled, when transition from C7816[TTYPE] = 0 to C7816[TTYPE] = 1 or when coming out of reset. In this case, it is assumed that the previous non-existent transfer was a received transfer.

The UART will automatically handle GT, CGT, and BGT such that the UART will not send a packet before the corresponding guard time expiring.

Table 49-356. Wait and guard time calculations

Parameter	Reset value [ETU]	C7816[TTYPE] = 0 [ETU]	C7816[TTYPE] = 1 [ETU]
Wait time (WT)	9600	$((WI + 1) \times 960 \times (GTFD + 1)) - 1$	Not used
Character wait time (CWT)	Not used	Not used	$11 + 2^{(CWI - 1)}$
Block wait time (BWT)	Not used	Not used	$10 + 2^{BWI} \times 960 \times (GTFD + 1)$
Guard time (GT)	12	**GTN not equal to 255** 12 + GTN **GTN equal to 255** 12	Not used
Character guard time (CGT)	Not used	Not used	**GTN not equal to 255** 12 + GTN **GTN equal to 255** 11
Block guard time (BGT)	Not used	Not used	22

49.4.8.5 Baud rate generation

The value in WF7816[GTFD] does not impact the clock frequency. SBR and BRFD are used to generate the clock frequency. This clock frequency is used by the UART only and is not seen by the smartcard device. The transmitter clocks operates at 1/16 the frequency of the receive clock so that the receiver is able to sample the received value 16 times during the ETU.

49.4.8.6 UART restrictions in ISO-7816 operation

Due to the flexibility of the UART module, there are several features and interrupts that are not supported while running in ISO-7816 mode. These restrictions are documented within the register field definitions.

49.4.9 Infrared interface

The UART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the UART. The IrDA physical layer specification defines a half-duplex infrared communication link for exchanging data. The full standard includes data rates up to 16 Mbits/s. This design covers data rates only between 2.4 kbits/s and 115.2 kbits/s.

The UART has an infrared transmit encoder and receive decoder. The UART transmits serial bits of data that are encoded by the infrared submodule to transmit a narrow pulse for every zero bit. No pulse is transmitted for every one bit. When receiving data, the IR pulses are detected using an IR photo diode and transformed to CMOS levels by the IR receive decoder, external from the MCU. The narrow pulses are then stretched by the infrared receive decoder to get back to a serial bit stream to be received by the UART. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active low pulses.

The infrared submodule receives its clock sources from the UART. One of these two clocks are selected in the infrared submodule to generate either 3/16, 1/16, 1/32, or 1/4 narrow pulses during transmission.

49.4.9.1 Infrared transmit encoder

The infrared transmit encoder converts serial bits of data from transmit shift register to the TXD signal. A narrow pulse is transmitted for a zero bit and no pulse for a one bit. The narrow pulse is sent in the middle of the bit with a duration of 1/32, 1/16, 3/16, or 1/4 of a bit time. A narrow high pulse is transmitted for a zero bit when C3[TXINV] is cleared, while a narrow low pulse is transmitted for a zero bit when C3[TXINV] is set.

49.4.9.2 Infrared receive decoder

The infrared receive block converts data from the RXD signal to the receive shift register. A narrow pulse is expected for each zero received and no pulse is expected for each one received. A narrow high pulse is expected for a zero bit when S2[RXINV] is cleared, while a narrow low pulse is expected for a zero bit when S2[RXINV] is set. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

49.5 Reset

All registers reset to a particular value are indicated in Memory map and registers.

49.6 System level interrupt sources

There are several interrupt signals that are sent from the UART. The following table lists the interrupt sources generated by the UART. The local enables for the UART interrupt sources are described in this table. Details regarding the individual operation of each interrupt are contained under various sub-sections of Memory map and registers. However, RXEDGIF description also outlines additional details regarding the RXEDGIF interrupt because of its complexity of operation. Any of the UART interrupt requests listed in the table can be used to bring the CPU out of Wait mode.

Table 49-357. UART interrupt sources

Interrupt Source	Flag	Local enable	DMA select
Transmitter	TDRE	TIE	TDMAS = 0
Transmitter	TC	TCIE	-
Receiver	IDLE	ILIE	-
Receiver	RDRF	RIE	RDMAS = 0
Receiver	LBKDIF	LBKDIE	-
Receiver	RXEDGIF	RXEDGIE	-
Receiver	OR	ORIE	-
Receiver	NF	NEIE	-
Receiver	FE	FEIE	-
Receiver	PF	PEIE	-
Receiver	RXUF	RXUFE	-
Transmitter	TXOF	TXOFE	-
Receiver	WT	WTWE	-
Receiver	CWT	CWTE	-

Table continues on the next page...

Table 49-357. UART interrupt sources (continued)

Interrupt Source	Flag	Local enable	DMA select
Receiver	BWT	BWTE	-
Receiver	INITD	INITDE	-
Receiver	TXT	TXTE	-
Receiver	RXT	RXTE	-
Receiver	GTV	GTVE	-

49.6.1 RXEDGIF description

S2[RXEDGIF] is set when an active edge is detected on the RxD pin. Therefore, the active edge can be detected only when in two wire mode. A RXEDGIF interrupt is generated only when S2[RXEDGIF] is set. If RXEDGIE is not enabled before S2[RXEDGIF] is set, an interrupt is not generated until S2[RXEDGIF] is set.

49.6.1.1 RxD edge detect sensitivity

Edge sensitivity can be software programmed to be either falling or rising. The polarity of the edge sensitivity is selected using S2[RXINV]. To detect the falling edge, S2[RXINV] is programmed to 0. To detect the rising edge, S2[RXINV] is programmed to 1.

Synchronizing logic is used prior to detect edges. Prior to detecting an edge, the receive data on RxD input must be at the deasserted logic level. A falling edge is detected when the RxD input signal is seen as a logic 1 (the deasserted level) during one module clock cycle, and then a logic 0 (the asserted level) during the next cycle. A rising edge is detected when the input is seen as a logic 0 during one module clock cycle and then a logic 1 during the next cycle.

49.6.1.2 Clearing RXEDGIF interrupt request

Writing a logic 1 to S2[RXEDGIF] immediately clears the RXEDGIF interrupt request even if the RxD input remains asserted. S2[RXEDGIF] remains set if another active edge is detected on RxD while attempting to clear S2[RXEDGIF] by writing a 1 to it.

49.6.1.3 Exit from low-power modes

The receive input active edge detect circuit is still active on low power modes (Wait and Stop). An active edge on the receive input brings the CPU out of low power mode if the interrupt is not masked (S2[RXEDGIF] = 1).

49.7 DMA operation

In the transmitter, S1[TDRE] can be configured to assert a DMA transfer request. In the receiver, S1[RDRF] can be configured to assert a DMA transfer request. The following table shows the configuration field settings required to configure each flag for DMA operation.

Table 49-358. DMA configuration

Flag	Request enable bit	DMA select bit
TDRE	TIE = 1	TDMAS = 1
RDRF	RIE = 1	RDMAS = 1

When a flag is configured for a DMA request, its associated DMA request is asserted when the flag is set. When S1[RDRF] is configured as a DMA request, the clearing mechanism of reading S1, followed by reading D, does not clear the associated flag. The DMA request remains asserted until an indication is received that the DMA transactions are done. When this indication is received, the flag bit and the associated DMA request is cleared. If the DMA operation failed to remove the situation that caused the DMA request, another request is issued.

49.8 Application information

This section describes the UART application information.

49.8.1 Transmit/receive data buffer operation

The UART has independent receive and transmit buffers. The size of these buffers may vary depending on the implementation of the module. The implemented size of the buffers is a fixed constant via PFIFO[TXFIFOSIZE] and PFIFO[RXFIFOSIZE]. Additionally, legacy support is provided that allows for the FIFO structure to operate as a

depth of one. This is the default/reset behavior of the module and can be adjusted using the PFIFO[RXFE] and PFIFO[TXFE] bits. Individual watermark levels are also provided for transmit and receive.

There are multiple ways to ensure that a data block, which is a set of characters, has completed transmission. These methods include:

1. Set TXFIFO[TXWATER] to 0. TDRE asserts when there is no further data in the transmit buffer. Alternatively the S1[TC] flag can be used to indicate when the transmit shift register is also empty.
2. Poll TCFIFO[TXCOUNT]. Assuming that only data for a data block has been put into the data buffer, when TCFIFO[TXCOUNT] = 0, all data has been transmitted or is in the process of transmission.
3. S1[TC] can be monitored. When S1[TC] asserts, it indicates that all data has been transmitted and there is no data currently being transmitted in the shift register.

49.8.2 ISO-7816 initialization sequence

This section outlines how to program the UART for ISO-7816 operation. Elements such as procedures to power up or power down the smartcard, and when to take those actions, are beyond the scope of this description. To set up the UART for ISO-7816 operation:

1. Select a baud rate. Write this value to the UART baud registers (BDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the BDH has no effect without also writing to BDL. According to the 7816 specification the initial (default) baud rating setting should be Fi = 372 and Di = 1 and a maximum frequency of 5 MHz. In other words, the BDH, BDL, and C4 registers should be programmed such that the transmission frequency provided to the smartcard device must be 1/372th of the clock and must not exceed 5 MHz.
2. Write to set BDH[LBKDIE] = 0.
3. Write to C1 to configure word length, parity, and other configuration fields (LOOPS, RSRC) and set C1[M] = 1, C1[PE] = 1, and C1[PT] = 0.
4. Write to set S2[RWUID] = 0 and S2[LBKDE] = 0.
5. Write to set MODEM[RXRTSE] = 0, MODEM[TXRTSPOL] = 0, MODEM[TXRTSE] = 0, and MODEM[TXCTSE] = 0.

6. Write to set up interrupt enable fields desired (C3[ORIE], C3[NEIE], C3[PEIE], and C3[FEIE])
7. Write to set C4[MAEN1] = 0 and C4[MAEN2] = 0.
8. Write to C5 register and configure DMA control register fields as desired for application.
9. Write to set C7816[INIT] = 1,C7816[TTYPE] = 0, and C7816[ISO_7816E] = 1. Program C7816[ONACK] and C7816[ANACK] as desired.
10. Write to IE7816 to set interrupt enable parameters as desired.
11. Write to ET7816 and set as desired.
12. Write to set C2[ILIE] = 0, C2[RE] = 1, C2[TE] = 1, C2[RWU] = 0, and C2[SBK] = 0. Set up interrupt enables C2[TIE], C2[TCIE], and C2[RIE] as desired.

At this time, the UART will start listening for an initial character. After being identified, it will automatically adjust S2[MSBF], C3[TXINV], and S2[RXINV]. The software must then receive and process an answer to reset. Upon processing the answer to reset, the software must write to set C2[RE] = 0 and C2[TE] = 0. The software should then adjust 7816 specific and UART generic parameters to match and configure data that was received during the answer on reset period. After the new settings have been programmed, including the new baud rate and C7816[TTYPE], C2[RE] and C2[TE] can be reenabled as required.

49.8.2.1 Transmission procedure for (C7816[TTYPE] = 0)

When the protocol selected is C7816[TTYPE] = 0, it is assumed that the software has a prior knowledge of who should be transmitting and receiving. Therefore, no mechanism is provided for automated transmission/receipt control. The software must monitor S1[TDRE], or configure for an interrupt, and provide additional data for transmission, as appropriate. Additionally, software should set C2[TE] = 1 and control TXDIR whenever it is the UART's turn to transmit information. For ease of monitoring, it is suggested that only data be transmitted until the next receiver/transmit switchover is loaded into the transmit FIFO/buffer.

49.8.2.2 Transmission procedure for (C7816[TTYPE] = 1)

When the protocol selected is C7816[TTYPE] = 1, data is transferred in blocks. Before starting a transmission, the software must write the size, in number of bytes, for the Information Field portion of the block into TLEN. If a CRC is being transmitted for the block, the value in TLEN must be one more than the size of the information field. The software must then set C2[TE] = 1 and C2[RE] = 1. The software must then monitor S1[TDRE]/interrupt and write the prologue, information, and epilogue field to the transmit buffer. TLEN automatically decrements, except for prologue bytes and the final epilogue byte. When the final epilogue byte has been transmitted, the UART automatically clears C2[TE] and C3[TXDIR] to 0, and the UART automatically starts capturing the response to the block that was transmitted. After the software has detected the receipt of the response, the transmission process must be repeated as needed with sufficient urgency to ensure that the block wait time and character wait times are not violated.

49.8.3 Initialization sequence (non ISO-7816)

To initiate a UART transmission:

1. Configure the UART.
   a. Select a baud rate. Write this value to the UART baud registers (BDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the BDH has no effect without also writing to BDL.
   b. Write to C1 to configure word length, parity, and other configuration bits (LOOPS, RSRC, M, WAKE, ILT, PE, and PT). Write to C4, MA1, and MA2 to configure.
   c. Enable the transmitter, interrupts, receiver, and wakeup as required, by writing to C2 (TIE, TCIE, RIE, ILIE, TE, RE, RWU, and SBK), S2 (MSBF and BRK13), and C3 (ORIE, NEIE, PEIE, and FEIE). A preamble or idle character is then shifted out of the transmitter shift register.
2. Transmit procedure for each byte.
   a. Monitor S1[TDRE] by reading S1 or responding to the TDRE interrupt. The amount of free space in the transmit buffer directly using TCFIFO[TXCOUNT] can also be monitored.

b. If the TDRE flag is set, or there is space in the transmit buffer, write the data to be transmitted to (C3[T8]/D). A new transmission will not result until data exists in the transmit buffer.

3. Repeat step 2 for each subsequent transmission.

Note

During normal operation, S1[TDRE] is set when the shift register is loaded with the next data to be transmitted from the transmit buffer and the number of datawords contained in the transmit buffer is less than or equal to the value in TWFIFO[TXWATER]. This occurs 9/16ths of a bit time after the start of the stop bit of the previous frame.

To separate messages with preambles with minimum idle line time, use this sequence between messages.

1. Write the last dataword of the first message to C3[T8]/D.
2. Wait for S1[TDRE] to go high with TWFIFO[TXWATER] = 0, indicating the transfer of the last frame to the transmit shift register.
3. Queue a preamble by clearing and then setting C2[TE].
4. Write the first and subsequent datawords of the second message to C3[T8]/D.

49.8.4 Overrun (OR) flag implications

To be flexible, the overrun flag (OR) operates slight differently depending on the mode of operation. There may be implications that need to be carefully considered. This section clarifies the behavior and the resulting implications. Regardless of mode, if a dataword is received while S1[OR] is set, S1[RDRF] and S1[IDLE] are blocked from asserting. If S1[RDRF] or S1[IDLE] were previously asserted, they will remain asserted until cleared.

49.8.4.1 Overrun operation

The assertion of S1[OR] indicates that a significant event has occurred. The assertion indicates that received data has been lost because there was a lack of room to store it in the data buffer. Therefore, while S1[OR] is set, no further data is stored in the data buffer until S1[OR] is cleared. This ensures that the application will be able to handle the overrun condition.

In most applications, because the total amount of lost data is known, the application will attempt to return the system to a known state. Before S1[OR] is cleared, all received data will be dropped. For this, the software does the following.

1. Remove data from the receive data buffer. This could be done by reading data from the data buffer and processing it if the data in the FIFO was still valuable when the overrun event occurred, or using CFIFO[RXFLUSH] to clear the buffer.
2. Clear S1[OR]. Note that if data was cleared using CFIFO[RXFLUSH], then clearing S1[OR] will result in SFIFO[RXUF] asserting. This is because the only way to clear S1[OR] requires reading additional information from the FIFO. Care should be taken to disable the SFIFO[RXUF] interrupt prior to clearing the OR flag and then clearing SFIFO[RXUF] after the OR flag has been cleared.

Note that, in some applications, if an overrun event is responded to fast enough, the lost data can be recovered. For example, when C7816[ISO_7816E] is asserted, C7816[TTYPE]=1 and C7816[ONACK] = 1, the application may reasonably be able to determine whether the lost data will be resent by the device. In this scenario, flushing the receiver data buffer may not be required. Rather, if S1[OR] is cleared, the lost data may be resent and therefore may be recoverable.

When LIN break detect (LBKDE) is asserted, S1[OR] has significantly different behavior than in other modes. S1[OR] will be set, regardless of how much space is actually available in the data buffer, if a LIN break character has been detected and the corresponding flag, S2[LBKDIF], is not cleared before the first data character is received after S2[LBKDIF] asserted. This behavior is intended to allow the software sufficient time to read the LIN break character from the data buffer to ensure that a break character was actually detected. The checking of the break character was used on some older implementations and is therefore supported for legacy reasons. Applications that do not require this checking can simply clear S2[LBKDIF] without checking the stored value to ensure it is a break character.

49.8.5 Overrun NACK considerations

When C7816[ISO_7816E] is enabled and C7816[TTYPE] = 0, the retransmission feature of the 7816 protocol can be used to help avoid lost data when the data buffer overflows. Using C7816[ONACK], the module can be programmed to issue a NACK on an overflow event. Assuming that the smartcard device has implemented retransmission, the lost data will be retransmitted. While useful, there is a programming implication that may require special consideration. The need to transmit a NACK must be determined and committed to prior to the dataword being fully received. While the NACK is being received, it is possible that the application code will read the data buffer such that

sufficient room will be made to store the dataword that is being NACKed. Even if room has been made in the data buffer after the transmission of a NACK is completed, the received data will always be discarded as a result of an overflow and the ET7816[RXTHRESHOLD] value will be incremented by one. However, if sufficient space now exists to write the received data which was NACK'ed, S1[OR] will be blocked and kept from asserting.

49.8.6 Match address registers

The two match address registers allow a second match address function for a broadcast or general call address to the serial bus, as an example.

49.8.7 Modem feature

This section describes the modem features.

49.8.7.1 Ready-to-receive using RTS

To help to stop overrun of the receiver data buffer, the RTS signal can be used by the receiver to indicate to another UART that it is ready to receive data. The other UART can send the data when its CTS signal is asserted. This handshaking conforms to the TIA-232-E standard. A transceiver is necessary if the required voltage levels of the communication link do not match the voltage levels of the UART's RTS and CTS signals.

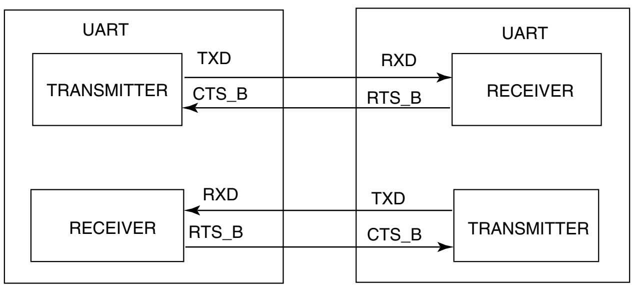


Figure 49-362. Ready-to-receive

The transmitter's CTS signal can be used for hardware flow control whether its RTS signal is used for hardware flow control, transceiver driver enable, or not at all.

49.8.7.2 Transceiver driver enable using RTS

RS-485 is a multiple drop communication protocol in which the UART transceiver's driver is 3-stated unless the UART is driving. The RTS signal can be used by the transmitter to enable the driver of a transceiver. The polarity of RTS can be matched to the polarity of the transceiver's driver enable signal. See the following figure.

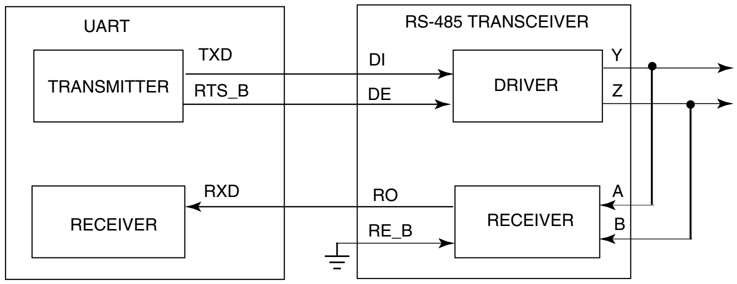


Figure 49-363. Transceiver driver enable using RTS

In the figure, the receiver enable signal is asserted. Another option for this connection is to connect RTS_B to both DE and RE_B. The transceiver's receiver is disabled while driving. A pullup can pull RXD to a non-floating value during this time. This option can be refined further by operating the UART in single wire mode, freeing the RXD pin for other uses.

49.8.8 IrDA minimum pulse width

The IrDA specifies a minimum pulse width of 1.6 µs. The UART hardware does not include a mechanism to restrict/force the pulse width to be greater than or equal to 1.6 µs. However, configuring the baud rate to 115.2 kbit/s and the narrow pulse width to 3/16 of a bit time results in a pulse width of 1.6 µs.

49.8.9 Clearing 7816 wait timer (WT, BWT, CWT) interrupts

The 7816 wait timer interrupts associated with IS7816[WT], IS7816[BWT], and IS7816[CWT] will automatically reassert if they are cleared and the wait time is still violated. This behavior is similar to most of the other interrupts on the UART. In most cases, if the condition that caused the interrupt to trigger still exists when the interrupt is cleared, then the interrupt will reassert. For example, consider the following scenario:

1. IS7816[WT] is programmed to assert after 9600 cycles of unresponsiveness.
2. The 9600 cycles pass without a response resulting in the WT interrupt asserting.
3. The IS7816[WT] is cleared at cycle 9700 by the interrupt service routine.

4. After the WT interrupt has been cleared, the smartcard remains unresponsive. At cycle 9701 the WT interrupt will be reasserted.

If the intent of clearing the interrupt is such that it does not reassert, the interrupt service routine must remove or clear the condition that originally caused the interrupt to assert prior to clearing the interrupt. There are multiple ways that this can be accomplished, including ensuring that an event that results in the wait timer resetting occurs, such as, the transmission of another packet.

49.8.10 Legacy and reverse compatibility considerations

Recent versions of the UART have added several new features. Whenever reasonably possible, reverse compatibility was maintained. However, in some cases this was either not feasible or the behavior was deemed as not intended. This section describes several differences to legacy operation that resulted from these recent enhancements. If application code from previous versions is used, it must be reviewed and modified to take the following items into account. Depending on the application code, additional items that are not listed here may also need to be considered.

1. Various reserved registers and register bits are used, such as, MSFB and M10.
2. This module now generates an error when invalid address spaces are used.
3. While documentation indicated otherwise, in some cases it was possible for S1[IDLE] to assert even if S1[OR] was set.
4. S1[OR] will be set only if the data buffer (FIFO) does not have sufficient room. Previously, the data buffer was always a fixed size of one and the S1[OR] flag would set so long as S1[RDRF] was set even if there was room in the data buffer. While the clearing mechanism has remained the same for S1[RDRF], keeping the OR flag assertion tied to the RDRF event rather than the data buffer being full would have greatly reduced the usefulness of the buffer when its size is larger than one.
5. Previously, when C2[RWU] was set (and WAKE = 0), the IDLE flag could reassert up to every bit period causing an interrupt and requiring the host processor to reassert C2[RWU]. This behavior has been modified. Now, when C2[RWU] is set (and WAKE = 0), at least one non-idle bit must be detected before an idle can be detected.

Chapter 50
Secured digital host controller (SDHC)

50.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The chapter is intended for a module driver software developer. It describes module-level operation and programming.

50.2 Overview

50.2.1 Supported types of cards

Different types of cards supported by the SDHC are described briefly as follows:

The multimedia card (MMC) is a universal low-cost data storage and communication media that is designed to cover a wide area of applications including mobile video and gaming. Old MMC cards are based on a 7-pin serial bus with a single data pin, while the new high-speed MMC communication is based on an advanced 11-pin serial bus designed to operate in the low-voltage range.

The secure digital card (SD) is an evolution of the old MMC technology. It is specifically designed to meet the security, capacity, performance, and environment requirements inherent in newly emerging audio and video consumer electronic devices. The physical form factor, pin assignment, and data transfer protocol are forward compatible with the old MMC with some additions.

Under the SD protocol, it can be categorized into memory card, I/O card and combo card, which has both memory and I/O functions. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard. The I/O card, which is also known as SDIO card, provides high-speed data I/O with low power consumption for mobile electronic devices. For the sake of simplicity, the figure does not show cards with reduced size or mini cards.

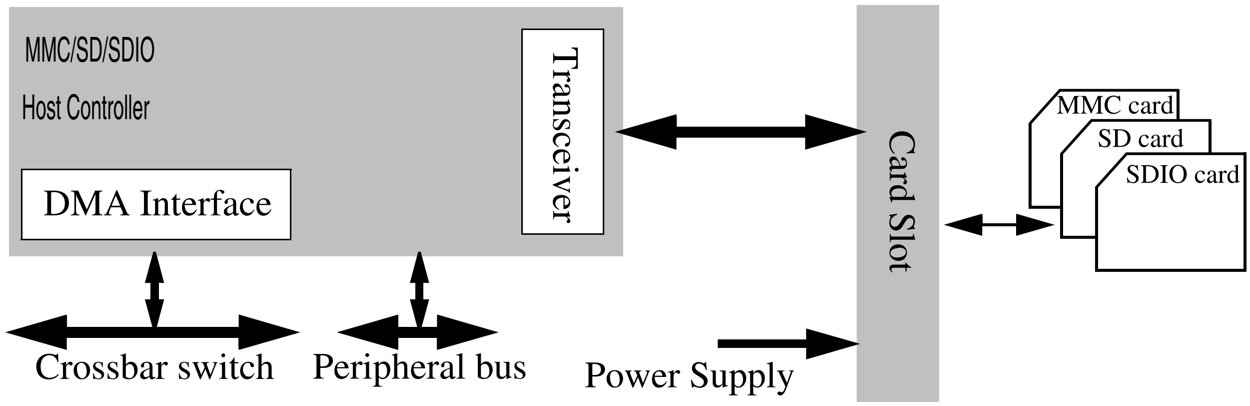


Figure 50-1. System connection of the SDHC

50.2.2 SDHC block diagram

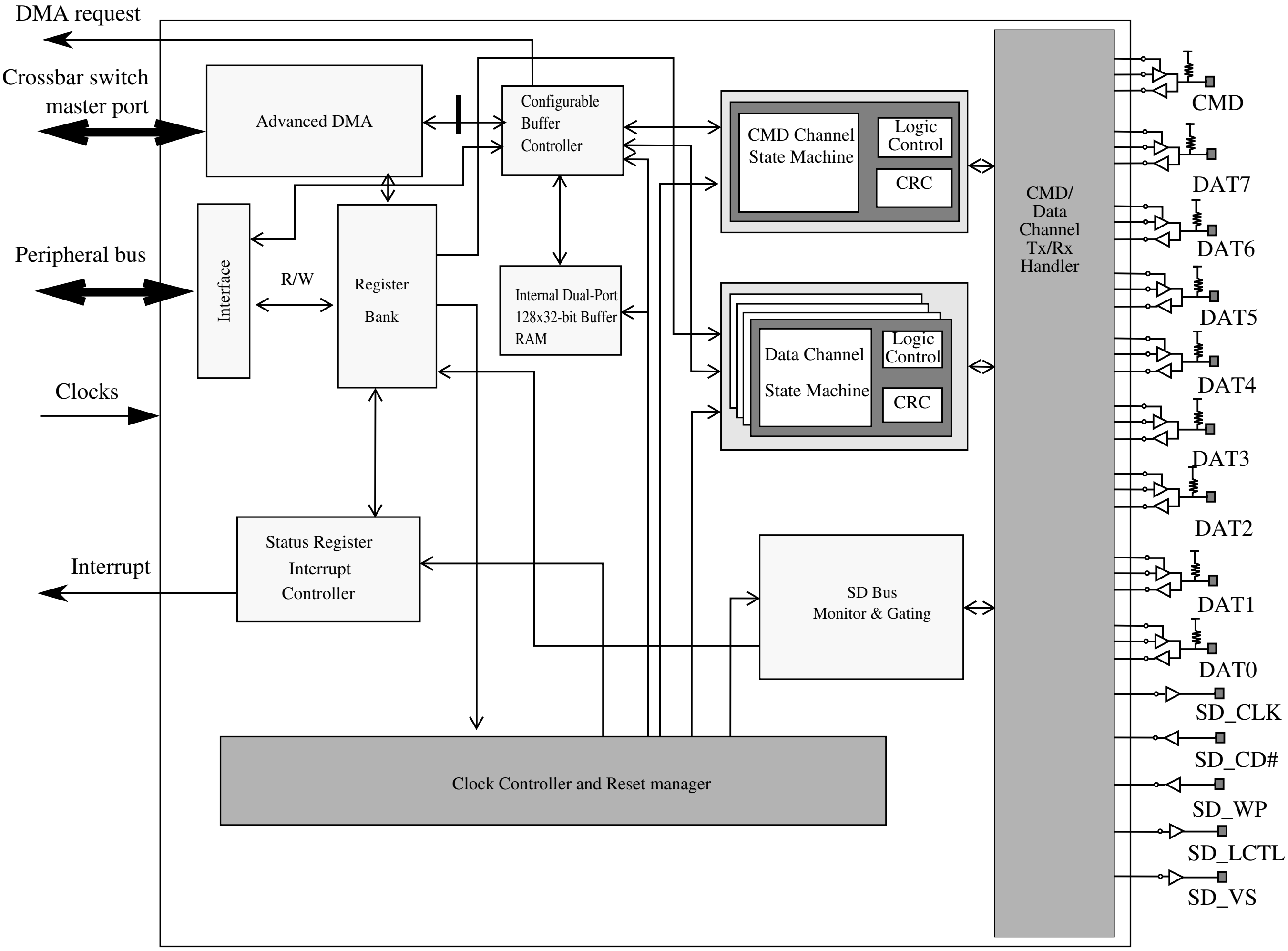


Figure 50-2. Enhanced secure digital host controller block diagram

50.2.3 Features

The features of the SDHC module include:

- Conforms to the SD Host Controller Standard Specification version 2.0 including test event register support
- Compatible with the MMC System Specification version 4.2/4.3
- Compatible with the SD Memory Card Specification version 2.0 and supports the high capacity SD memory card
- Compatible with the SDIO Card Specification version 2.0

- Designed to work with SD memory, miniSD memory, SDIO, miniSDIO, SD Combo, MMC, MMC plus, and MMC RS cards
- Card bus clock frequency up to 52 MHz
- Supports 1-bit/4-bit SD and SDIO modes, 1-bit/4-bit / 8-bit MMC modesdevices
    - Up to 200 Mbps of data transfer for SD/SDIO cards using 4 parallel data lines
    - Up to 416 Mbps of data transfer for MMC cards using 8 parallel data lines in Single Data Rate (SDR) mode
- Supports single block, multiblock read and write
- Supports block sizes of 1 ~ 4096 bytes
- Supports the write protection switch for write operations
- Supports both synchronous and asynchronous abort (both hardware and software CMD12)
- Supports pause during the data transfer at block gap
- Supports SDIO Read Wait and Suspend Resume operations
- Supports auto CMD12 for multiblock transfer
- Host can initiate non-data transfer command while data transfer is in progress
- Allows cards to interrupt the host in 1-bit and 4-bit SDIO modes, also supports interrupt period
- Embodies a fully configurable 128x32-bit FIFO for read/write data
- Supports internal DMA capabilities
- Supports advanced DMA to perform linked memory access

50.2.4 Modes and operations

The SDHC can select the following modes for data transfer:

- SD 1-bit
- SD 4-bit
- MMC 1-bit
- MMC 4-bit

- MMC 8-bit
- Identification mode up to 400 kHz
- MMC Full Speed mode up to 20 MHz
- MMC High Speed mode up to 52 MHz
- SD/SDIO Full Speed mode up to 25 MHz
- SD/SDIO High Speed mode up to 50 MHz

50.3 SDHC signal descriptions

Table 50-1. SDHC signal descriptions

Signal	Description	I/O
SDHC_DCLK	Generated clock used to drive the MMC, SD, SDIO or CE-ATA cards.	O
SDHC_CMD	Send commands to and receive responses from the card.	I/O
SDHC_D0	DAT0 line or busy-state detect	I/O
SDHC_D1	8-bit mode: DAT1 line 4-bit mode: DAT1 line or interrupt detect 1-bit mode: Interrupt detect	I/O
SDHC_D2	4-/8-bit mode: DAT2 line or read wait 1-bit mode: Read wait	I/O
SDHC_D3	4-/8-bit mode: DAT3 line or configured as card detection pin 1-bit mode: May be configured as card detection pin	I/O
SDHC_D4	DAT4 line in 8-bit mode Not used in other modes	I/O
SDHC_D5	DAT5 line in 8-bit mode Not used in other modes	I/O
SDHC_D6	DAT6 line in 8-bit mode Not used in other modes	I/O
SDHC_D7	DAT7 line in 8-bit mode Not used in other modes	I/O

50.4 Memory map and register definition

This section includes the module memory map and detailed descriptions of all registers.

SDHC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_1000	DMA System Address register (SDHC0_DSADDR)	32	R/W	0000_0000h	50.4.1/2742
400B_1004	Block Attributes register (SDHC0_BLKATTR)	32	R/W	0000_0000h	50.4.2/2742
400B_1008	Command Argument register (SDHC0_CMDARG)	32	R/W	0000_0000h	50.4.3/2744
400B_100C	Transfer Type register (SDHC0_XFERTYP)	32	R/W	0000_0000h	50.4.4/2744
400B_1010	Command Response 0 (SDHC0_CMDRSP0)	32	R	0000_0000h	50.4.5/2748
400B_1014	Command Response 1 (SDHC0_CMDRSP1)	32	R	0000_0000h	50.4.6/2749
400B_1018	Command Response 2 (SDHC0_CMDRSP2)	32	R	0000_0000h	50.4.7/2749
400B_101C	Command Response 3 (SDHC0_CMDRSP3)	32	R	0000_0000h	50.4.8/2749
400B_1020	Buffer Data Port register (SDHC0_DATPORT)	32	R/W	0000_0000h	50.4.9/2751
400B_1024	Present State register (SDHC0_PRSSTAT)	32	R	0000_0000h	50.4.10/ 2751
400B_1028	Protocol Control register (SDHC0_PROCTL)	32	R/W	0000_0020h	50.4.11/ 2757
400B_102C	System Control register (SDHC0_SYSCTL)	32	R/W	0000_8008h	50.4.12/ 2761
400B_1030	Interrupt Status register (SDHC0_IRQSTAT)	32	R/W	0000_0000h	50.4.13/ 2764
400B_1034	Interrupt Status Enable register (SDHC0_IRQSTATEN)	32	R/W	117F_013Fh	50.4.14/ 2769
400B_1038	Interrupt Signal Enable register (SDHC0_IRQSIGEN)	32	R/W	0000_0000h	50.4.15/ 2772
400B_103C	Auto CMD12 Error Status Register (SDHC0_AC12ERR)	32	R	0000_0000h	50.4.16/ 2774
400B_1040	Host Controller Capabilities (SDHC0_HTCAPBLT)	32	R	07F3_0000h	50.4.17/ 2778
400B_1044	Watermark Level register (SDHC0_WML)	32	R/W	0810_0810h	50.4.18/ 2780
400B_1050	Force Event register (SDHC0_FEVT)	32	W (always reads 0)	0000_0000h	50.4.19/ 2781
400B_1054	ADMA Error Status register (SDHC0_ADMAES)	32	R	0000_0000h	50.4.20/ 2783
400B_1058	ADMA System Addressregister (SDHC0_ADSADDR)	32	R/W	0000_0000h	50.4.21/ 2786
400B_10C0	Vendor Specific register (SDHC0_VENDOR)	32	R/W	0000_0001h	50.4.22/ 2787

Table continues on the next page...

SDHC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_10C4	MMC Boot register (SDHC0_MMCBOOT)	32	R/W	0000_0000h	50.4.23/ 2789
400B_10FC	Host Controller Version (SDHC0_HOSTVER)	32	R	0000_1201h	50.4.24/ 2790
400B_2000	DMA System Address register (SDHC1_DSADDR)	32	R/W	0000_0000h	50.4.1/2742
400B_2004	Block Attributes register (SDHC1_BLKATTR)	32	R/W	0000_0000h	50.4.2/2742
400B_2008	Command Argument register (SDHC1_CMDARG)	32	R/W	0000_0000h	50.4.3/2744
400B_200C	Transfer Type register (SDHC1_XFERTYP)	32	R/W	0000_0000h	50.4.4/2744
400B_2010	Command Response 0 (SDHC1_CMDRSP0)	32	R	0000_0000h	50.4.5/2748
400B_2014	Command Response 1 (SDHC1_CMDRSP1)	32	R	0000_0000h	50.4.6/2749
400B_2018	Command Response 2 (SDHC1_CMDRSP2)	32	R	0000_0000h	50.4.7/2749
400B_201C	Command Response 3 (SDHC1_CMDRSP3)	32	R	0000_0000h	50.4.8/2749
400B_2020	Buffer Data Port register (SDHC1_DATPORT)	32	R/W	0000_0000h	50.4.9/2751
400B_2024	Present State register (SDHC1_PRSSTAT)	32	R	0000_0000h	50.4.10/ 2751
400B_2028	Protocol Control register (SDHC1_PROCTL)	32	R/W	0000_0020h	50.4.11/ 2757
400B_202C	System Control register (SDHC1_SYSCTL)	32	R/W	0000_8008h	50.4.12/ 2761
400B_2030	Interrupt Status register (SDHC1_IRQSTAT)	32	R/W	0000_0000h	50.4.13/ 2764
400B_2034	Interrupt Status Enable register (SDHC1_IRQSTATEN)	32	R/W	117F_013Fh	50.4.14/ 2769
400B_2038	Interrupt Signal Enable register (SDHC1_IRQSIGEN)	32	R/W	0000_0000h	50.4.15/ 2772
400B_203C	Auto CMD12 Error Status Register (SDHC1_AC12ERR)	32	R	0000_0000h	50.4.16/ 2774
400B_2040	Host Controller Capabilities (SDHC1_HTCAPBLT)	32	R	07F3_0000h	50.4.17/ 2778
400B_2044	Watermark Level register (SDHC1_WML)	32	R/W	0810_0810h	50.4.18/ 2780
400B_2050	Force Event register (SDHC1_FEVT)	32	W (always reads 0)	0000_0000h	50.4.19/ 2781
400B_2054	ADMA Error Status register (SDHC1_ADMAES)	32	R	0000_0000h	50.4.20/ 2783
400B_2058	ADMA System Addressregister (SDHC1_ADSADDR)	32	R/W	0000_0000h	50.4.21/ 2786
400B_20C0	Vendor Specific register (SDHC1_VENDOR)	32	R/W	0000_0001h	50.4.22/ 2787
400B_20C4	MMC Boot register (SDHC1_MMCBOOT)	32	R/W	0000_0000h	50.4.23/ 2789

Table continues on the next page...

SDHC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_20FC	Host Controller Version (SDHC1_HOSTVER)	32	R	0000_1201h	50.4.24/ 2790

50.4.1 DMA System Address register (SDHCx_DSADDR)

This register contains the physical system memory address used for DMA transfers.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSADDR																														0	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_DSADDR field descriptions

Field	Description
31–2 DSADDR	DMA System Address Contains the 32-bit system memory address for a DMA transfer. Because the address must be word (4 bytes) align, the least 2 bits are reserved, always 0. When the SDHC stops a DMA transfer, this register points to the system address of the next contiguous data position. It can be accessed only when no transaction is executing, that is, after a transaction has stopped. Read operation during transfers may return an invalid value. The host driver shall initialize this register before starting a DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. This register is protected during a data transfer. When data lines are active, write to this register is ignored. The host driver shall wait, until PRSSTAT[DLA] is cleared, before writing to this register. The SDHC internal DMA does not support a virtual memory system. It supports only continuous physical memory access. And due to AHB burst limitations, if the burst must cross the 1 KB boundary, SDHC will automatically change SEQ burst type to NSEQ. Because this register supports dynamic address reflecting, when IRQSTAT[TC] bit is set, it automatically alters the value of internal address counter, so SW cannot change this register when IRQSTAT[TC] is set.
1–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

50.4.2 Block Attributes register (SDHCx_BLKATTR)

This register is used to configure the number of data blocks and the number of bytes in each block.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BLKCNT																0			BLKSIZE												
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_BLKATTR field descriptions

Field	Description
31–16 BLKCNT	Blocks Count For Current Transfer This register is enabled when XFERTYP[BCEN] is set to 1 and is valid only for multiple block transfers. For single block transfer, this register will always read as 1. The host driver shall set this register to a value between 1 and the maximum block count. The SDHC decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to 0 results in no data blocks being transferred. This register must be accessed only when no transaction is executing, that is, after transactions are stopped. During data transfer, read operations on this register may return an invalid value and write operations are ignored. When saving transfer content as a result of a suspend command, the number of blocks yet to be transferred can be determined by reading this register. The reading of this register must be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend. This is because when suspend command is sent out, SDHC will regard the current transfer as aborted and change BLKCNT back to its original value instead of keeping the dynamical indicator of remained block count. When restoring transfer content prior to issuing a resume command, the host driver shall restore the previously saved block count. **NOTE:** Although the BLKCNT field is 0 after reset, the read of reset value is 0x1. This is because when XFERTYP[MSBSEL] is 0, indicating a single block transfer, the read value of BLKCNT is always 1. 0000h Stop count. 0001h 1 block 0002h 2 blocks ... FFFFh 65535 blocks
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–0 BLKSIZE	Transfer Block Size Specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing, that is, after a transaction has stopped. Read operations during transfers may return an invalid value, and write operations will be ignored. 000h No data transfer. 001h 1 Byte 002h 2 Bytes 003h 3 Bytes 004h 4 Bytes ... 1FFh 511 Bytes 200h 512 Bytes ... 800h 2048 Bytes ... 1000h 4096 Bytes

50.4.3 Command Argument register (SDHCx_CMDARG)

This register contains the SD/MMC command argument.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMDARG																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_CMDARG field descriptions

Field	Description
31–0 CMDARG	Command Argument The SD/MMC command argument is specified as bits 39-8 of the command format in the SD or MMC specification. This register is write protected when PRSSTAT[CDIHB0] is set.

50.4.4 Transfer Type register (SDHCx_XFERTYP)

This register is used to control the operation of data transfers. The host driver shall set this register before issuing a command followed by a data transfer, or before issuing a resume command. To prevent data loss, the SDHC prevents writing to the bits that are involved in the data transfer of this register, when data transfer is active. These bits are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

The host driver shall check PRSSTAT[CDIHB] and PRSSTAT[CIHB] before writing to this register. When PRSSTAT[CDIHB] is set, any attempt to send a command with data by writing to this register is ignored; when PRSSTAT[CIHB] bit is set, any write to this register is ignored.

On sending commands with data transfer involved, it is mandatory that the block size is nonzero. Besides, block count must also be nonzero, or indicated as single block transfer (bit 5 of this register is 0 when written), or block count is disabled (bit 1 of this register is 0 when written), otherwise SDHC will ignore the sending of this command and do nothing. For write command, with all above restrictions, it is also mandatory that the write protect switch is not active (WPSPL bit of Present State Register is 1), otherwise SDHC will also ignore the command.

If the commands with data transfer does not receive the response in 64 clock cycles, that is, response time-out, SDHC will regard the external device does not accept the command and abort the data transfer. In this scenario, the driver must issue the command again to retry the transfer. It is also possible that, for some reason, the card responds to the

command but SDHC does not receive the response, and if it is internal DMA (either simple DMA or ADMA) read operation, the external system memory is over-written by the internal DMA with data sent back from the card.

The following table shows the summary of how register settings determine the type of data transfer.

Table 50-40. Transfer Type register setting for various transfer types

Multi/Single block select	Block count enable	Block count	Function
0	Don't care	Don't care	Single transfer
1	0	Don't care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

The following table shows the relationship between XFERTYP[CICEN] and XFERTYP[CCCEN], in regards to XFERTYP[RSPTYP] as well as the name of the response type.

Table 50-41. Relationship between parameters and the name of the response type

Response type (RSPTYP)	Index check enable (CICEN)	CRC check enable (CCCEN)	Name of response type
00	0	0	No Response
01	0	1	IR2
10	0	0	R3,R4
10	1	1	R1,R5,R6
11	1	1	R1b,R5b

NOTE

- In the SDIO specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO specification. But R5b is defined in this specification to specify that the SDHC will check the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command shall be used with R5b.
- The CRC field for R3 and R4 is expected to be all 1 bits. The CRC check shall be disabled for these response types.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		CMDINX						CMDTYP		DPSEL	CICEN	CCCEN	0	RSPTYP	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										MSBSEL	DTDSEL	0	AC12EN	BCEN	DMAEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_XFERTYP field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–24 CMDINX	Command Index These bits shall be set to the command number that is specified in bits 45-40 of the command-format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.
23–22 CMDTYP	Command Type There are three types of special commands: suspend, resume, and abort. These bits shall be set to 00b for all other commands. • Suspend command: If the suspend command succeeds, the SDHC shall assume that the card bus has been released and that it is possible to issue the next command which uses the DAT line. Because the SDHC does not monitor the content of command response, it does not know if the suspend command succeeded or not. It is the host driver's responsibility to check the status of the suspend command and send another command marked as suspend to inform the SDHC that a suspend command was successfully issued. After the end bit of command is sent, the SDHC deasserts read wait for read transactions and stops checking busy for write transactions. In 4-bit mode, the interrupt cycle starts. If the suspend command fails, the SDHC will maintain its current state, and the host driver shall restart the transfer by setting PROCTL[CREQ]. • Resume command: The host driver restarts the data transfer by restoring the registers saved before sending the suspend command and then sends the resume command. The SDHC will check for a pending busy state before starting write transfers. • Abort command: If this command is set when executing a read transfer, the SDHC will stop reads to the buffer. If this command is set when executing a write transfer, the SDHC will stop driving the DAT line. After issuing the abort command, the host driver must issue a software reset (abort transaction). 00b Normal other commands. 01b Suspend CMD52 for writing bus suspend in CCCR. 10b Resume CMD52 for writing function select in CCCR. 11b Abort CMD12, CMD52 for writing I/O abort in CCCR.
21 DPSEL	Data Present Select This bit is set to 1 to indicate that data is present and shall be transferred using the DAT line. It is set to 0 for the following:

Table continues on the next page...

SDHCx_XFERTYP field descriptions (continued)

Field	Description
	• Commands using only the CMD line, for example: CMD52. • Commands with no data transfer, but using the busy signal on DAT[0] line, R1b or R5b, for example: CMD38. **NOTE:** In resume command, this bit shall be set, and other bits in this register shall be set the same as when the transfer was initially launched. When the Write Protect switch is on, that is, the WPSPL bit is active as 0, any command with a write operation will be ignored. That is to say, when this bit is set, while the DTDSEL bit is 0, writes to the register Transfer Type are ignored. 0b No data present. 1b Data present.
20 CICEN	Command Index Check Enable If this bit is set to 1, the SDHC will check the index field in the response to see if it has the same value as the command index. If it is not, it is reported as a command index error. If this bit is set to 0, the index field is not checked. 0b Disable 1b Enable
19 CCCEN	Command CRC Check Enable If this bit is set to 1, the SDHC shall check the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this bit is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. 0b Disable 1b Enable
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 RSPTYP	Response Type Select 00b No response. 01b Response length 136. 10b Response length 48. 11b Response length 48, check busy after response.
15–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 MSBSEL	Multi/Single Block Select Enables multiple block DAT line data transfers. For any other commands, this bit shall be set to 0. If this bit is 0, it is not necessary to set the block count register. 0b Single block. 1b Multiple blocks.
4 DTDSEL	Data Transfer Direction Select Defines the direction of DAT line data transfers. The bit is set to 1 by the host driver to transfer data from the SD card to the SDHC and is set to 0 for all other commands. 0b Write host to card. 1b Read card to host.

Table continues on the next page...

SDHCx_XFERTYP field descriptions (continued)

Field	Description
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 AC12EN	Auto CMD12 Enable Multiple block transfers for memory require a CMD12 to stop the transaction. When this bit is set to 1, the SDHC will issue a CMD12 automatically when the last block transfer has completed. The host driver shall not set this bit to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, the SDHC will ignore this bit whether it is set or not. 0b Disable 1b Enable
1 BCEN	Block Count Enable Used to enable the Block Count register, which is only relevant for multiple block transfers. When this bit is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 0b Disable 1b Enable
0 DMAEN	DMA Enable Enables DMA functionality. If this bit is set to 1, a DMA operation shall begin when the host driver sets the DPSEL bit of this register. Whether the simple DMA, or the advanced DMA, is active depends on PROCTL[DMAS]. 0b Disable 1b Enable

50.4.5 Command Response 0 (SDHCx_CMDRSP0)

This register is used to store part 0 of the response bits from the card.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP0																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_CMDRSP0 field descriptions

Field	Description
31–0 CMDRSP0	Command Response 0

50.4.6 Command Response 1 (SDHCx_CMDRSP1)

This register is used to store part 1 of the response bits from the card.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP1																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_CMDRSP1 field descriptions

Field	Description
31–0 CMDRSP1	Command Response 1

50.4.7 Command Response 2 (SDHCx_CMDRSP2)

This register is used to store part 2 of the response bits from the card.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP2																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_CMDRSP2 field descriptions

Field	Description
31–0 CMDRSP2	Command Response 2

50.4.8 Command Response 3 (SDHCx_CMDRSP3)

This register is used to store part 3 of the response bits from the card.

The following table describes the mapping of command responses from the SD bus to command response registers for each response type. In the table, R[ ] refers to a bit range within the response data as transmitted on the SD bus.

Table 50-46. Response bit definition for each response type

Response type	Meaning of response	Response field	Response register
R1,R1b (normal response)	Card status	R[39:8]	CMDRSP0
R1b (Auto CMD12 response)	Card status for auto CMD12	R[39:8]	CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{CMDRSP3[23:0], CMDRSP2, CMDRSP1, CMDRSP0}
R3 (OCR register)	OCR register for memory	R[39:8]	CMDRSP0
R4 (OCR register)	OCR register for I/O etc.	R[39:8]	CMDRSP0
R5, R5b	SDIO response	R[39:8]	CMDRSP0
R6 (Publish RCA)	New published RCA[31:16] and card status[15:0]	R[39:9]	CMDRSP0

This table shows that most responses with a length of 48 (R[47:0]) have 32-bit of the response data (R[39:8]) stored in the CMDRSP0 register. Responses of type R1b (auto CMD12 responses) have response data bits (R[39:8]) stored in the CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bit of the response data (R[127:8]) stored in the CMDRSP0, 1, 2, and 3 registers.

To be able to read the response status efficiently, the SDHC stores only a part of the response data in the command response registers. This enables the host driver to efficiently read 32-bit of response data in one read cycle on a 32-bit bus system. Parts of the response, the index field and the CRC, are checked by the SDHC, as specified by XFERTYP[CICEN] and XFERTYP[CCCEN], and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, the SDHC will check R[47:1], and if the response length is 136 the SDHC will check R[119:1].

Because the SDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, the SDHC stores the auto CMD12 response in the CMDRSP3 register. The CMD_wo_DAT response is stored in CMDRSP0. This allows the SDHC to avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When the SDHC modifies part of the command response registers, as shown in the table above, it preserves the unmodified bits.

Address: Base address + 1Ch offset

Bit	31–16	15–0
R	CMDRSP3	
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

SDHCx_CMDRSP3 field descriptions

Field	Description
31–0 CMDRSP3	Command Response 3

50.4.9 Buffer Data Port register (SDHCx_DATPORT)

This is a 32-bit data port register used to access the internal buffer and it cannot be updated in Idle mode.

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATCONT																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_DATPORT field descriptions

Field	Description
31–0 DATCONT	Data Content The Buffer Data Port register is for 32-bit data access by the CPU or the external DMA. When the internal DMA is enabled, any write to this register is ignored, and any read from this register will always yield 0s.

50.4.10 Present State register (SDHCx_PRSSTAT)

The host driver can get status of the SDHC from this 32-bit read-only register.

NOTE

The host driver can issue CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the DAT lines are busy during a data transfer. These commands can be issued when Command Inhibit (CIHB) is set to zero. Other commands shall be issued when Command Inhibit (CDIHB) is set to zero. Possible changes to the SD Physical Specification may add other commands to this list in the future.

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLSL								CLSL	0			WPSPL	CDPL	0	CINS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				BREN	BWEN	RTA	WTA	SDOFF	PEROFF	HCKOFF	IPGOFF	SDSTB	DLA	CDIHB	CIHB
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_PRSSTAT field descriptions

Field	Description
31–24 DLSL	DAT Line Signal Level Used to check the DAT line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DAT[0]. The reset value is effected by the external pullup/pulldown resistors. By default, the read value of this field after reset is 8'b11110111, when DAT[3] is pulled down and the other lines are pulled up. DAT[0] Data 0 line signal level. DAT[1] Data 1 line signal level. DAT[2] Data 2 line signal level. DAT[3] Data 3 line signal level. DAT[4] Data 4 line signal level. DAT[5] Data 5 line signal level. DAT[6] Data 6 line signal level. DAT[7] Data 7 line signal level.

Table continues on the next page...

SDHCx_PRSSTAT field descriptions (continued)

Field	Description
23 CLSL	CMD Line Signal Level Used to check the CMD line level to recover from errors, and for debugging. The reset value is effected by the external pullup/pulldown resistor, by default, the read value of this bit after reset is 1b, when the command line is pulled up.
22–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 WPSPL	Write Protect Switch Pin Level The write protect switch is supported for memory and combo cards.This bit reflects the inverted value of the SD_WP pin of the card socket. A software reset does not affect this bit. The reset value is effected by the external write protect switch. If the SD_WP pin is not used, it must be tied low, so that the reset value of this bit is high and write is enabled. 0b Write protected (SD_WP=1). 1b Write enabled (SD_WP=0).
18 CDPL	Card Detect Pin Level This bit reflects the inverse value of the SD_CD# pin for the card socket. Debouncing is not performed on this bit. This bit may be valid, but is not guaranteed, because of propagation delay. Use of this bit is limited to testing since it must be debounced by software. A software reset does not effect this bit. A write to the Force Event Register does not effect this bit. The reset value is effected by the external card detection pin. This bit shows the value on the SD_CD# pin (that is, when a card is inserted in the socket, it is 0 on the SD_CD# input, and consequently the CDPL reads 1.) 0b No card present (SD_CD#=1). 1b Card present (SD_CD#=0).
17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 CINS	Card Inserted Indicates whether a card has been inserted. The SDHC debounces this signal so that the host driver will not need to wait for it to stabilize. Changing from a 0 to 1 generates a card insertion interrupt in the Interrupt Status register. Changing from a 1 to 0 generates a card removal interrupt in the Interrupt Status register. A write to the force event register does not effect this bit. SYSCTL[RSTA] does not effect this bit. A software reset does not effect this bit. 0b Power on reset or no card. 1b Card inserted.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 BREN	Buffer Read Enable Used for non-DMA read transfers. The SDHC may implement multiple buffers to transfer data efficiently. This read-only flag indicates that valid data exists in the host side buffer. If this bit is high, valid data greater than the watermark level exist in the buffer. This read-only flag indicates that valid data exists in the host side buffer. 0b Read disable, valid data less than the watermark level exist in the buffer. 1b Read enable, valid data greater than the watermark level exist in the buffer.
10 BWEN	Buffer Write Enable

Table continues on the next page...

SDHCx_PRSSTAT field descriptions (continued)

Field	Description
	Used for non-DMA write transfers. The SDHC can implement multiple buffers to transfer data efficiently. This read-only flag indicates whether space is available for write data. If this bit is 1, valid data greater than the watermark level can be written to the buffer. This read-only flag indicates whether space is available for write data. 0b Write disable, the buffer can hold valid data less than the write watermark level. 1b Write enable, the buffer can hold valid data greater than the write watermark level.
9 RTA	Read Transfer Active Used for detecting completion of a read transfer. This bit is set for either of the following conditions: • After the end bit of the read command. • When writing a 1 to PROCTL[CREQ] to restart a read transfer. A transfer complete interrupt is generated when this bit changes to 0. This bit is cleared for either of the following conditions: • When the last data block as specified by block length is transferred to the system, that is, all data are read away from SDHC internal buffer. • When all valid data blocks have been transferred from SDHC internal buffer to the system and no current block transfers are being sent as a result of the stop at block gap request being set to 1. 0b No valid data. 1b Transferring data.
8 WTA	Write Transfer Active Indicates that a write transfer is active. If this bit is 0, it means no valid write data exists in the SDHC. This bit is set in either of the following cases: • After the end bit of the write command. • When writing 1 to PROCTL[CREQ] to restart a write transfer. This bit is cleared in either of the following cases: • After getting the CRC status of the last data block as specified by the transfer count (single and multiple). • After getting the CRC status of any block where data transmission is about to be stopped by a stop at block gap request. During a write transaction, a block gap event interrupt is generated when this bit is changed to 0, as result of the stop at block gap request being set. This status is useful for the host driver in determining when to issue commands during write busy state. 0b No valid data. 1b Transferring data.
7 SDOFF	SD Clock Gated Off Internally Indicates that the SD clock is internally gated off, because of buffer over/under-run or read pause without read wait assertion, or the driver has cleared SYSCTL[SDCLKEN] to stop the SD clock. This bit is for the host driver to debug data transaction on the SD bus. 0b SD clock is active. 1b SD clock is gated off.
6 PEROFF	SDHC clock Gated Off Internally Indicates that the is internally gated off. This bit is for the host driver to debug transaction on the SD bus. When INITA bit is set, SDHC sending 80 clock cycles to the card, SDCLKEN must be 1 to enable the

Table continues on the next page...

SDHCx_PRSSTAT field descriptions (continued)

Field	Description
	output card clock, otherwise the will never be gate off, so and will be always active. SDHC clockSDHC clockSDHC clockbus clock 0b SDHC clock is active. 1b SDHC clock is gated off.
5 HCKOFF	System Clock Gated Off Internally Indicates that the system clock is internally gated off. This bit is for the host driver to debug during a data transfer. 0b System clock is active. 1b System clock is gated off.
4 IPGOFF	Bus Clock Gated Off Internally Indicates that the bus clock is internally gated off. This bit is for the host driver to debug. 0b Bus clock is active. 1b Bus clock is gated off.
3 SDSTB	SD Clock Stable Indicates that the internal card clock is stable. This bit is for the host driver to poll clock status when changing the clock frequency. It is recommended to clear SYSCTL[SDCLKEN] to remove glitch on the card clock when the frequency is changing. 0b Clock is changing frequency and not stable. 1b Clock is stable.
2 DLA	Data Line Active Indicates whether one of the DAT lines on the SD bus is in use. **In the case of read transactions:** This status indicates whether a read transfer is executing on the SD bus. Changes in this value from 1 to 0, between data blocks, generates a block gap event interrupt in the Interrupt Status register. This bit will be set in either of the following cases: • After the end bit of the read command. • When writing a 1 to PROCTL[CREQ] to restart a read transfer. This bit will be cleared in either of the following cases: 1. When the end bit of the last data block is sent from the SD bus to the SDHC. 2. When the read wait state is stopped by a suspend command and the DAT2 line is released. The SDHC will wait at the next block gap by driving read wait at the start of the interrupt cycle. If the read wait signal is already driven (data buffer cannot receive data), the SDHC can wait for a current block gap by continuing to drive the read wait signal. It is necessary to support read wait to use the suspend / resume function. This bit will remain 1 during read wait.

Table continues on the next page...

SDHCx_PRSSTAT field descriptions (continued)

Field	Description
	In the case of write transactions: This status indicates that a write transfer is executing on the SD bus. Changes in this value from 1 to 0 generate a transfer complete interrupt in the interrupt status register. This bit will be set in either of the following cases: • After the end bit of the write command. • When writing to 1 to PROCTL[CREQ] to continue a write transfer. This bit will be cleared in either of the following cases: • When the SD card releases write busy of the last data block, the SDHC will also detect if the output is not busy. If the SD card does not drive the busy signal after the CRC status is received, the SDHC shall assume the card drive "Not busy". • When the SD card releases write busy, prior to waiting for write transfer, and as a result of a stop at block gap request. **In the case of command with busy pending:** This status indicates that a busy state follows the command and the data line is in use. This bit will be cleared when the DAT0 line is released. 0b DAT line inactive. 1b DAT line active.
1 CDIHB	Command Inhibit (DAT) This status bit is generated if either the DLA or the RTA is set to 1. If this bit is 0, it indicates that the SDHC can issue the next SD/MMC Command. Commands with a busy signal belong to CDIHB, for example, R1b, R5b type. Except in the case when the command busy is finished, changing from 1 to 0 generates a transfer complete interrupt in the Interrupt Status register. **NOTE:** The SD host driver can save registers for a suspend transaction after this bit has changed from 1 to 0. 0b Can issue command which uses the DAT line. 1b Cannot issue command which uses the DAT line.
0 CIHB	Command Inhibit (CMD) If this status bit is 0, it indicates that the CMD line is not in use and the SDHC can issue a SD/MMC Command using the CMD line. This bit is set also immediately after the Transfer Type register is written. This bit is cleared when the command response is received. Even if the CDIHB bit is set to 1, Commands using only the CMD line can be issued if this bit is 0. Changing from 1 to 0 generates a command complete interrupt in the interrupt status register. If the SDHC cannot issue the command because of a command conflict error (see command CRC error) or because of a command not issued by auto CMD12 error, this bit will remain 1 and the command complete is not set. The status of issuing an auto CMD12 does not show on this bit. 0b Can issue command using only CMD line. 1b Cannot issue command.

50.4.11 Protocol Control register (SDHCx_PROCTL)

There are three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether the SDHC issues a suspend command or the SD card accepts the suspend command:

- If the host driver does not issue a suspend command, the continue request shall be used to restart the transfer.
- If the host driver issues a suspend command and the SD card accepts it, a resume command shall be used to restart the transfer.
- If the host driver issues a suspend command and the SD card does not accept it, the continue request shall be used to restart the transfer.

Any time stop at block gap request stops the data transfer, the host driver shall wait for a transfer complete (in the interrupt status register), before attempting to restart the transfer. When restarting the data transfer by continue request, the host driver shall clear the stop at block gap request before or simultaneously.

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					WECRM	WECINS	WECINT	0				IABG	RWCTL	CREQ	SABGREQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						DMAS		CDSS	CDTL	EMODE		D3CD	DTW		LCTL
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

SDHCx_PROCTL field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 WECRM	Wakeup Event Enable On SD Card Removal Enables a wakeup event, via IRQSTAT[CRM]. FN_WUS (Wake Up Support) in CIS does not effect this bit. When this bit is set, IRQSTAT[CRM] and the SDHC interrupt can be asserted without SD_CLK toggling. When the wakeup feature is not enabled, the SD_CLK must be active to assert IRQSTAT[CRM] and the SDHC interrupt.

Table continues on the next page...

SDHCx_PROCTL field descriptions (continued)

Field	Description
	0b Disabled 1b Enabled
25 WECINS	Wakeup Event Enable On SD Card Insertion Enables a wakeup event, via IRQSTAT[CINS]. FN_WUS (Wake Up Support) in CIS does not effect this bit. When this bit is set, IRQSTATEN[CINSEN] and the SDHC interrupt can be asserted without SD_CLK toggling. When the wakeup feature is not enabled, the SD_CLK must be active to assert IRQSTATEN[CINSEN] and the SDHC interrupt. 0b Disabled 1b Enabled
24 WECINT	Wakeup Event Enable On Card Interrupt Enables a wakeup event, via IRQSTAT[CINT]. This bit can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1. When this bit is set, the card interrupt status and the SDHC interrupt can be asserted without SD_CLK toggling. When the wakeup feature is not enabled, the SD_CLK must be active to assert the card interrupt status and the SDHC interrupt. 0b Disabled 1b Enabled
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 IABG	Interrupt At Block Gap Valid only in 4-bit mode, of the SDIO card, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SDIO card can't signal an interrupt during a multiple block transfer, this bit must be set to 0 to avoid an inadvertent interrupt. When the host driver detects an SDIO card insertion, it shall set this bit according to the CCCR of the card. 0b Disabled 1b Enabled
18 RWCTL	Read Wait Control The read wait function is optional for SDIO cards. If the card supports read wait, set this bit to enable use of the read wait protocol to stop read data using the DAT[2] line. Otherwise, the SDHC has to stop the SD Clock to hold read data, which restricts commands generation. When the host driver detects an SDIO card insertion, it shall set this bit according to the CCCR of the card. If the card does not support read wait, this bit shall never be set to 1, otherwise DAT line conflicts may occur. If this bit is set to 0, stop at block gap during read operation is also supported, but the SDHC will stop the SD Clock to pause reading operation. 0b Disable read wait control, and stop SD clock at block gap when SABGREQ is set. 1b Enable read wait control, and assert read wait without stopping SD clock at block gap when SABGREQ bit is set.
17 CREQ	Continue Request Used to restart a transaction which was stopped using the PROCTL[SABGREQ]. When a suspend operation is not accepted by the card, it is also by setting this bit to restart the paused transfer. To cancel stop at the block gap, set PROCTL[SABGREQ] to 0 and set this bit to 1 to restart the transfer. The SDHC automatically clears this bit, therefore it is not necessary for the host driver to set this bit to 0. If both PROCTL[SABGREQ] and this bit are 1, the continue request is ignored.

Table continues on the next page...

SDHCx_PROCTL field descriptions (continued)

Field	Description
	0b No effect. 1b Restart
16 SABGREQ	Stop At Block Gap Request Used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the IRQSTATEN[TCSEN] is set to 1, indicating a transfer completion, the host driver shall leave this bit set to 1. Clearing both PROCTL[SABGREQ] and PROCTL[CREQ] does not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The SDHC will honor the PROCTL[SABGREQ] for write transfers, but for read transfers it requires that SDIO card support read wait. Therefore, the host driver shall not set this bit during read transfers unless the SDIO card supports read wait and has set PROCTL[RWCTL] to 1, otherwise the SDHC will stop the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the host driver writes data to the data port register, the host driver shall set this bit after all block data is written. If this bit is set to 1, the host driver shall not write data to the Data Port register after a block is sent. Once this bit is set, the host driver shall not clear this bit before IRQSTATEN[TCSEN] is set, otherwise the SDHC's behavior is undefined. This bit effects PRSSTAT[RTA], PRSSTAT[WTA], and PRSSTAT[CDIHB]. 0b Transfer 1b Stop
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 DMAS	DMA Select This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation. 00 No DMA or simple DMA is selected. 01 ADMA1 is selected. 10 ADMA2 is selected. 11 Reserved
7 CDSS	Card Detect Signal Selection Selects the source for the card detection. 0b Card detection level is selected for normal purpose. 1b Card detection test level is selected for test purpose.
6 CDTL	Card Detect Test Level Enabled while the CDSS is set to 1 and it indicates card insertion. 0b Card detect test level is 0, no card inserted. 1b Card detect test level is 1, card inserted.
5–4 EMODE	Endian Mode The SDHC supports all four endian modes in data transfer. 00b Big endian mode 01b Half word big endian mode 10b Little endian mode 11b Reserved

Table continues on the next page...

SDHCx_PROCTL field descriptions (continued)

Field	Description
3 D3CD	DAT3 As Card Detection Pin If this bit is set, DAT3 should be pulled down to act as a card detection pin. Be cautious when using this feature, because DAT3 is also a chip-select for the SPI mode. A pulldown on this pin and CMD0 may set the card into the SPI mode, which the SDHC does not support. Note: Keep this bit set if SDIO interrupt is used. 0b DAT3 does not monitor card Insertion. 1b DAT3 as card detection pin.
2–1 DTW	Data Transfer Width Selects the data width of the SD bus for a data transfer. The host driver shall set it to match the data width of the card. Possible data transfer width is 1-bit, 4-bits or 8-bits. 00b 1-bit mode 01b 4-bit mode 10b 8-bit mode 11b Reserved
0 LCTL	LED Control This bit, fully controlled by the host driver, is used to caution the user not to remove the card while the card is being accessed. If the software is going to issue multiple SD commands, this bit can be set during all these transactions. It is not necessary to change for each transaction. When the software issues multiple SD commands, setting the bit once before the first command is sufficient: it is not necessary to reset the bit between commands. 0b LED off. 1b LED on.

50.4.12 System Control register (SDHCx_SYSCTL)

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				INITA	0	0	0	0				DTOCV			
W						RSTD	RSTC	RSTA								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SDCLKFS								DVS				SDCLKEN	PEREN	HCKEN	IPGEN
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

SDHCx_SYSCTL field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 INITA	Initialization Active When this bit is set, 80 SD-clocks are sent to the card. After the 80 clocks are sent, this bit is self-cleared. This bit is very useful during the card power-up period when 74 SD-clocks are needed and the clock auto gating feature is enabled. Writing 1 to this bit when this bit is already 1 has no effect. Writing 0 to this bit at any time has no effect. When either of the PRSSTAT[CIHB] and PRSSTAT[CDIHB] bits are set, writing 1 to this bit is ignored, that is, when command line or data lines are active, write to this bit is not allowed. On the otherhand, when this bit is set, that is, during intialization active period, it is allowed to issue command, and the command bit stream will appear on the CMD pad after all 80 clock cycles are done. So when this command ends, the driver can make sure the 80 clock cycles are sent out. This is very useful when the driver needs send 80 cycles to the card and does not want to wait till this bit is self-cleared.
26 RSTD	Software Reset For DAT Line Only part of the data circuit is reset. DMA circuit is also reset. The following registers and bits are cleared by this bit: • Data Port register • Buffer Is Cleared And Initialized.Present State register • Buffer Read Enable • Buffer Write Enable • Read Transfer Active

Table continues on the next page...

SDHCx_SYSCTL field descriptions (continued)

Field	Description
	• Write Transfer Active • DAT Line Active • Command Inhibit (DAT) Protocol Control register • Continue Request • Stop At Block Gap Request Interrupt Status register • Buffer Read Ready • Buffer Write Ready • DMA Interrupt • Block Gap Event • Transfer Complete 0b No reset. 1b Reset.
25 RSTC	Software Reset For CMD Line Only part of the command circuit is reset. The following registers and bits are cleared by this bit: • PRSSTAT[CIHB] • IRQSTAT[CC] 0b No reset. 1b Reset.
24 RSTA	Software Reset For ALL Effects the entire host controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared. During its initialization, the host driver shall set this bit to 1 to reset the SDHC. The SDHC shall reset this bit to 0 when the capabilities registers are valid and the host driver can read them. Additional use of software reset for all does not affect the value of the capabilities registers. After this bit is set, it is recommended that the host driver reset the external card and reinitialize it. 0b No reset. 1b Reset.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 DTOCV	Data Timeout Counter Value Determines the interval by which DAT line timeouts are detected. See IRQSTAT[DTOE] for information on factors that dictate time-out generation. Time-out clock frequency will be generated by dividing the base clock SDCLK value by this value. The host driver can clear IRQSTATEN[DTOESEN] to prevent inadvertent time-out events. 0000b SDCLK x 2^{13} 0001b SDCLK x 2^{14} ... 1110b SDCLK x 2^{27} 1111b Reserved
15–8 SDCLKFS	SDCLK Frequency Select Used to select the frequency of the SDCLK pin. The frequency is not programmed directly. Rather this register holds the prescaler (this register) and divisor (next register) of the base clock frequency register. Setting 00h bypasses the frequency prescaler of the SD Clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of SDHC clock and the following divisor bits.

Table continues on the next page...

SDHCx_SYSCTL field descriptions (continued)

Field	Description
	The frequency of SDCLK is set by the following formula: Clock frequency = (Base clock) / (prescaler x divisor) For example, if the base clock frequency is 96 MHz, and the target frequency is 25 MHz, then choosing the prescaler value of 01h and divisor value of 1h will yield 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 kHz, the prescaler value of 08h and divisor value of eh yields the exact clock value of 400 kHz. The reset value of this field is 80h, so if the input base clock (SDHC clock) is about 96 MHz, the default SD clock after reset is 375 kHz. According to the SD Physical Specification Version 1.1 and the SDIO Card Specification Version 1.2, the maximum SD clock frequency is 50 MHz and shall never exceed this limit. Only the following settings are allowed: 01h Base clock divided by 2. 02h Base clock divided by 4. 04h Base clock divided by 8. 08h Base clock divided by 16. 10h Base clock divided by 32. 20h Base clock divided by 64. 40h Base clock divided by 128. 80h Base clock divided by 256.
7–4 DVS	Divisor Used to provide a more exact divisor to generate the desired SD clock frequency. Note the divider can even support odd divisor without deterioration of duty cycle. The setting are as following: 0h Divisor by 1. 1h Divisor by 2. ... Eh Divisor by 15. Fh Divisor by 16.
3 SDCLKEN	SD Clock Enable The host controller shall stop SDCLK when writing this bit to 0. SDCLK frequency can be changed when this bit is 0. Then, the host controller shall maintain the same clock frequency until SDCLK is stopped (stop at SDCLK = 0). If the IRQSTAT[CINS] is cleared, this bit must be cleared by the host driver to save power.
2 PEREN	Peripheral Clock Enable If this bit is set, SDHC clock will always be active and no automatic gating is applied. Thus the SDCLK is active except for when auto gating-off during buffer danger (buffer about to over-run or under-run). When this bit is cleared, the SDHC clock will be automatically off whenever there is no transaction on the SD bus. Because this bit is only a feature enabling bit, clearing this bit does not stop SDCLK immediately. The SDHC clock will be internally gated off, if none of the following factors are met: • The cmd part is reset, or • Data part is reset, or • A soft reset, or • The cmd is about to send, or • Clock divisor is just updated, or • Continue request is just set, or • This bit is set, or • Card insertion is detected, or • Card removal is detected, or

Table continues on the next page...

SDHCx_SYSCTL field descriptions (continued)

Field	Description
	• Card external interrupt is detected, or • 80 clocks for initialization phase is ongoing 0b SDHC clock will be internally gated off. 1b SDHC clock will not be automatically gated off.
1 HCKEN	System Clock Enable If this bit is set, system clock will always be active and no automatic gating is applied. When this bit is cleared, system clock will be automatically off when no data transfer is on the SD bus. 0b System clock will be internally gated off. 1b System clock will not be automatically gated off.
0 IPGEN	IPG Clock Enable If this bit is set, bus clock will always be active and no automatic gating is applied. The bus clock will be internally gated off, if none of the following factors are met: • The cmd part is reset, or • Data part is reset, or • Soft reset, or • The cmd is about to send, or • Clock divisor is just updated, or • Continue request is just set, or • This bit is set, or • Card insertion is detected, or • Card removal is detected, or • Card external interrupt is detected, or • The SDHC clock is not gated off **NOTE:** The bus clock will not be auto gated off if the SDHC clock is not gated off. So clearing only this bit has no effect unless the PEREN bit is also cleared. 0b Bus clock will be internally gated off. 1b Bus clock will not be automatically gated off.

50.4.13 Interrupt Status register (SDHCx_IRQSTAT)

An interrupt is generated when the Normal Interrupt Signal Enable is enabled and at least one of the status bits is set to 1. For all bits, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. For Card Interrupt, before writing 1 to clear, it is required that the card stops asserting the interrupt, meaning that when the Card Driver services the interrupt condition, otherwise the CINT bit will be asserted again.

The table below shows the relationship between the CTOE and the CC bits.

Table 50-52. SDHC status for CTOE/CC bit combinations

Command complete	Command timeout error	Meaning of the status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

The table below shows the relationship between the Transfer Complete and the Data Timeout Error.

Table 50-53. SDHC status for data timeout error/transfer complete bit combinations

Transfer complete	Data timeout error	Meaning of the status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data transfer complete

The table below shows the relationship between the command CRC Error (CCE) and Command Timeout Error (CTOE).

Table 50-54. SDHC status for CCE/CTOE Bit Combinations

Command complete	Command timeout error	Meaning of the status
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DMAE	0			AC12E	0	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
W				w1c				w1c		w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CINT	CRM	CINS	BRR	BWR	DINT	BGE	TC	CC
W								w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_IRQSTAT field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 DMAE	DMA Error Occurs when an Internal DMA transfer has failed. This bit is set to 1, when some error occurs in the data transfer. This error can be caused by either Simple DMA or ADMA, depending on which DMA is in use. The value in DMA System Address register is the next fetch address where the error occurs. Because any error corrupts the whole data block, the host driver shall restart the transfer from the corrupted block boundary. The address of the block boundary can be calculated either from the current DSADDR value or from the remaining number of blocks and the block size. 0b No error. 1b Error.
27–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 AC12E	Auto CMD12 Error Occurs when detecting that one of the bits in the Auto CMD12 Error Status register has changed from 0 to 1. This bit is set to 1, not only when the errors in Auto CMD12 occur, but also when the Auto CMD12 is not executed due to the previous command error. 0b No error. 1b Error.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 DEBE	Data End Bit Error Occurs either when detecting 0 at the end bit position of read data, which uses the DAT line, or at the end bit position of the CRC. 0b No error. 1b Error.
21 DCE	Data CRC Error Occurs when detecting a CRC error when transferring read data, which uses the DAT line, or when detecting the Write CRC status having a value other than 010. 0b No error. 1b Error.

Table continues on the next page...

SDHCx_IRQSTAT field descriptions (continued)

Field	Description
20 DTOE	Data Timeout Error Occurs when detecting one of following time-out conditions. • Busy time-out for R1b,R5b type • Busy time-out after Write CRC status • Read Data time-out 0b No error. 1b Time out.
19 CIE	Command Index Error Occurs if a Command Index error occurs in the command response. 0b No error. 1b Error.
18 CEBE	Command End Bit Error Occurs when detecting that the end bit of a command response is 0. 0b No error. 1b End Bit Error generated.
17 CCE	Command CRC Error Command CRC Error is generated in two cases. • If a response is returned and the Command Timeout Error is set to 0, indicating no time-out, this bit is set when detecting a CRC error in the command response. • The SDHC detects a CMD line conflict by monitoring the CMD line when a command is issued. If the SDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then the SDHC shall abort the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error shall also be set to 1 to distinguish CMD line conflict. 0b No error. 1b CRC Error generated.
16 CTOE	Command Timeout Error Occurs only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the SDHC detects a CMD line conflict, in which case a Command CRC Error shall also be set, this bit shall be set without waiting for 64 SDCLK cycles. This is because the command will be aborted by the SDHC. 0b No error. 1b Time out.
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 CINT	Card Interrupt This status bit is set when an interrupt signal is detected from the external card. In 1-bit mode, the SDHC will detect the Card Interrupt without the SD Clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so the interrupt from card can only be sampled during interrupt cycle, introducing some delay between the interrupt signal from the SDIO card and the interrupt to the host system. Writing this bit to 1 can clear this bit, but as the interrupt factor from the SDIO card does not clear, this bit is set again. To clear this bit, it is required to reset the interrupt factor from the external card followed by a writing 1 to this bit.

Table continues on the next page...

SDHCx_IRQSTAT field descriptions (continued)

Field	Description
	When this status has been set, and the host driver needs to service this interrupt, the Card Interrupt Signal Enable in the Interrupt Signal Enable register should be 0 to stop driving the interrupt signal to the host system. After completion of the card interrupt service (it must reset the interrupt factors in the SDIO card and the interrupt signal may not be asserted), write 1 to clear this bit, set the Card Interrupt Signal Enable to 1, and start sampling the interrupt signal again. 0b No Card Interrupt. 1b Generate Card Interrupt.
7 CRM	Card Removal This status bit is set if the Card Inserted bit in the Present State register changes from 1 to 0. When the host driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register must be confirmed. Because the card state may possibly be changed when the host driver clears this bit and the interrupt event may not be generated. When this bit is cleared, it will be set again if no card is inserted. To leave it cleared, clear the Card Removal Status Enable bit in Interrupt Status Enable register. 0b Card state unstable or inserted. 1b Card removed.
6 CINS	Card Insertion This status bit is set if the Card Inserted bit in the Present State register changes from 0 to 1. When the host driver writes this bit to 1 to clear this status, the status of the Card Inserted in the Present State register must be confirmed. Because the card state may possibly be changed when the host driver clears this bit and the interrupt event may not be generated. When this bit is cleared, it will be set again if a card is inserted. To leave it cleared, clear the Card Inserted Status Enable bit in Interrupt Status Enable register. 0b Card state unstable or removed. 1b Card inserted.
5 BRR	Buffer Read Ready This status bit is set if the Buffer Read Enable bit, in the Present State register, changes from 0 to 1. See the Buffer Read Enable bit in the Present State register for additional information. 0b Not ready to read buffer. 1b Ready to read buffer.
4 BWR	Buffer Write Ready This status bit is set if the Buffer Write Enable bit, in the Present State register, changes from 0 to 1. See the Buffer Write Enable bit in the Present State register for additional information. 0b Not ready to write buffer. 1b Ready to write buffer.
3 DINT	DMA Interrupt Occurs only when the internal DMA finishes the data transfer successfully. Whenever errors occur during data transfer, this bit will not be set. Instead, the DMAE bit will be set. Either Simple DMA or ADMA finishes data transferring, this bit will be set. 0b No DMA Interrupt. 1b DMA Interrupt is generated.

Table continues on the next page...

SDHCx_IRQSTAT field descriptions (continued)

Field	Description
2 BGE	Block Gap Event If PROCTL[SABGREQ] is set, this bit is set when a read or write transaction is stopped at a block gap. If PROCTL[SABGREQ] is not set to 1, this bit is not set to 1. In the case of a read transaction: This bit is set at the falling edge of the DAT line active status, when the transaction is stopped at SD Bus timing. The read wait must be supported in order to use this function. In the case of write transaction: This bit is set at the falling edge of write transfer active status, after getting CRC status at SD bus timing. 0b No block gap event. 1b Transaction stopped at block gap.
1 TC	Transfer Complete This bit is set when a read or write transfer is completed. In the case of a read transaction: This bit is set at the falling edge of the read transfer active status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by the data length, after the last data has been read to the host system. The second is when data has stopped at the block gap and completed the data transfer by setting PROCTL[SABGREQ], after valid data has been read to the host system. In the case of a write transaction: This bit is set at the falling edge of the DAT line active status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by the data length and the busy signal is released. The second is when data transfers are stopped at the block gap, by setting PROCTL[SABGREQ], and the data transfers are completed,after valid data is written to the SD card and the busy signal released. 0b Transfer not complete. 1b Transfer complete.
0 CC	Command Complete This bit is set when you receive the end bit of the command response, except Auto CMD12. See PRSSTAT[CIHB]. 0b Command not complete. 1b Command complete.

50.4.14 Interrupt Status Enable register (SDHCx_IRQSTATEN)

Setting the bits in this register to 1 enables the corresponding interrupt status to be set by the specified event. If any bit is cleared, the corresponding interrupt status bit is also cleared, that is, when the bit in this register is cleared, the corresponding bit in interrupt status register is always 0.

NOTE

- Depending on PROCTL[IABG] bit setting, SDHC may be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There will be some delays on the card interrupt, asserted from the card, to the time the host system is informed.
- To detect a CMD line conflict, the host driver must set both IRQSTATEN[CTOESEN] and IRQSTATEN[CCESEN] to 1.

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DMAESEN	0			AC12ESEN	0	DEBESEN	DCESEN	DTOESEN	CIESEN	CEBESEN	CCESEN	CTOESEN
W																
Reset	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CINTSEN	CRMSEN	CINSEN	BRRSEN	BWRSEN	DINTSEN	BGESEN	TCSEN	CCSEN
W																
Reset	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1

SDHCx_IRQSTATEN field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 DMAESEN	DMA Error Status Enable 0b Masked 1b Enabled
27–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 AC12ESEN	Auto CMD12 Error Status Enable 0b Masked 1b Enabled
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 DEBESEN	Data End Bit Error Status Enable 0b Masked 1b Enabled

Table continues on the next page...

SDHCx_IRQSTATEN field descriptions (continued)

Field	Description
21 DCESEN	Data CRC Error Status Enable 0b Masked 1b Enabled
20 DTOESEN	Data Timeout Error Status Enable 0b Masked 1b Enabled
19 CIESEN	Command Index Error Status Enable 0b Masked 1b Enabled
18 CEBESEN	Command End Bit Error Status Enable 0b Masked 1b Enabled
17 CCESEN	Command CRC Error Status Enable 0b Masked 1b Enabled
16 CTOESEN	Command Timeout Error Status Enable 0b Masked 1b Enabled
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 CINTSEN	Card Interrupt Status Enable If this bit is set to 0, the SDHC will clear the interrupt request to the system. The card interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The host driver must clear the this bit before servicing the card interrupt and must set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 0b Masked 1b Enabled
7 CRMSEN	Card Removal Status Enable 0b Masked 1b Enabled
6 CINSEN	Card Insertion Status Enable 0b Masked 1b Enabled
5 BRRSEN	Buffer Read Ready Status Enable 0b Masked 1b Enabled
4 BWRSEN	Buffer Write Ready Status Enable

Table continues on the next page...

SDHCx_IRQSTATEN field descriptions (continued)

Field	Description
	0b Masked 1b Enabled
3 DINTSEN	DMA Interrupt Status Enable 0b Masked 1b Enabled
2 BGESEN	Block Gap Event Status Enable 0b Masked 1b Enabled
1 TCSEN	Transfer Complete Status Enable 0b Masked 1b Enabled
0 CCSEN	Command Complete Status Enable 0b Masked 1b Enabled

50.4.15 Interrupt Signal Enable register (SDHCx_IRQSIGEN)

This register is used to select which interrupt status is indicated to the host system as the interrupt. All of these status bits share the same interrupt line. Setting any of these bits to 1 enables interrupt generation. The corresponding status register bit will generate an interrupt when the corresponding interrupt signal enable bit is set.

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			DMAEIEN	0			AC12EIEN	0	DEBEIEN	DCEIEN	DTOEIEN	CIEIEN	CEBEIEN	CCEIEN	CTOEIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							CINTIEN	CRMIEN	CINSIEN	BRRIEN	BWRIEN	DINTIEN	BGEIEN	TCIEN	CCIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_IRQSIGEN field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 DMAEIEN	DMA Error Interrupt Enable 0b Masked 1b Enabled
27–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 AC12EIEN	Auto CMD12 Error Interrupt Enable 0b Masked 1b Enabled
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 DEBEIEN	Data End Bit Error Interrupt Enable 0b Masked 1b Enabled
21 DCEIEN	Data CRC Error Interrupt Enable 0b Masked 1b Enabled
20 DTOEIEN	Data Timeout Error Interrupt Enable 0b Masked 1b Enabled
19 CIEIEN	Command Index Error Interrupt Enable 0b Masked 1b Enabled
18 CEBEIEN	Command End Bit Error Interrupt Enable 0b Masked 1b Enabled
17 CCEIEN	Command CRC Error Interrupt Enable 0b Masked 1b Enabled
16 CTOEIEN	Command Timeout Error Interrupt Enable 0b Masked 1b Enabled
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 CINTIEN	Card Interrupt Enable 0b Masked 1b Enabled

Table continues on the next page...

SDHCx_IRQSIGEN field descriptions (continued)

Field	Description
7 CRMIEN	Card Removal Interrupt Enable 0b Masked 1b Enabled
6 CINSIEN	Card Insertion Interrupt Enable 0b Masked 1b Enabled
5 BRRIEN	Buffer Read Ready Interrupt Enable 0b Masked 1b Enabled
4 BWRIEN	Buffer Write Ready Interrupt Enable 0b Masked 1b Enabled
3 DINTIEN	DMA Interrupt Enable 0b Masked 1b Enabled
2 BGEIEN	Block Gap Event Interrupt Enable 0b Masked 1b Enabled
1 TCIEN	Transfer Complete Interrupt Enable 0b Masked 1b Enabled
0 CCIEN	Command Complete Interrupt Enable 0b Masked 1b Enabled

50.4.16 Auto CMD12 Error Status Register (SDHCx_AC12ERR)

When the AC12ESEN bit in the Status register is set, the host driver shall check this register to identify what kind of error the Auto CMD12 indicated. This register is valid only when the Auto CMD12 Error status bit is set.

The following table shows the relationship between the Auto CMGD12 CRC error and the Auto CMD12 command timeout error.

Table 50-58. Relationship between Command CRC Error and Command Timeout Error For Auto CMD12

Auto CMD12 CRC error	Auto CMD12 timeout error	Type of error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Changes in Auto CMD12 Error Status register can be classified in three scenarios:

1. When the SDHC is going to issue an Auto CMD12:
    - Set bit 0 to 1 if the Auto CMD12 can't be issued due to an error in the previous command.
    - Set bit 0 to 0 if the auto CMD12 is issued.
2. At the end bit of an auto CMD12 response:
    - Check errors corresponding to bits 1-4.
    - Set bits 1-4 corresponding to detected errors.
    - Clear bits 1-4 corresponding to detected errors.
3. Before reading the Auto CMD12 error status bit 7:
    - Set bit 7 to 1 if there is a command that can't be issued.
    - Clear bit 7 if there is no command to issue.

The timing for generating the auto CMD12 error and writing to the command register are asynchronous. After that, bit 7 shall be sampled when the driver is not writing to the command register. So it is suggested to read this register only when IRQSTAT[AC12E] is set. An Auto CMD12 error interrupt is generated when one of the error bits (0-4) is set to 1. The command not issued by auto CMD12 error does not generate an interrupt.

Address: Base address + 3Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CNIBAC12E	0		AC12IE	AC12CE	AC12EBE	AC12TOE	AC12NE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_AC12ERR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CNIBAC12E	Command Not Issued By Auto CMD12 Error Setting this bit to 1 means CMD_wo_DAT is not executed due to an auto CMD12 error (D04-D01) in this register. 0b No error. 1b Not issued.
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 AC12IE	Auto CMD12 Index Error Occurs if the command index error occurs in response to a command. 0b No error. 1b Error, the CMD index in response is not CMD12.

Table continues on the next page...

SDHCx_AC12ERR field descriptions (continued)

Field	Description
3 AC12CE	Auto CMD12 CRC Error Occurs when detecting a CRC error in the command response. 0b No CRC error. 1b CRC error met in Auto CMD12 response.
2 AC12EBE	Auto CMD12 End Bit Error Occurs when detecting that the end bit of command response is 0 which must be 1. 0b No error. 1b End bit error generated.
1 AC12TOE	Auto CMD12 Timeout Error Occurs if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to 1, the other error status bits (2-4) have no meaning. 0b No error. 1b Time out.
0 AC12NE	Auto CMD12 Not Executed If memory multiple block data transfer is not started, due to a command error, this bit is not set because it is not necessary to issue an auto CMD12. Setting this bit to 1 means the SDHC cannot issue the auto CMD12 to stop a memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (1-4) have no meaning. 0b Executed. 1b Not executed.

50.4.17 Host Controller Capabilities (SDHCx_HTCAPBLT)

This register provides the host driver with information specific to the SDHC implementation. The value in this register is the power-on-reset value, and does not change with a software reset. Any write to this register is ignored.

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					Reserved	Reserved	VS33	SRS	DMAS	HSS	ADMAS	0	MBL		
W						Reserved	Reserved									
Reset	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_HTCAPBLT field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 Reserved	This field is reserved.
25 Reserved	This field is reserved.
24 VS33	Voltage Support 3.3 V This bit shall depend on the host system ability. 0b 3.3 V not supported. 1b 3.3 V supported.
23 SRS	Suspend/Resume Support This bit indicates whether the SDHC supports suspend / resume functionality. If this bit is 0, the suspend and resume mechanism, as well as the read Wwait, are not supported, and the host driver shall not issue either suspend or resume commands. 0b Not supported. 1b Supported.
22 DMAS	DMA Support This bit indicates whether the SDHC is capable of using the internal DMA to transfer data between system memory and the data buffer directly. 0b DMA not supported. 1b DMA supported.
21 HSS	High Speed Support

Table continues on the next page...

SDHCx_HTCAPBLT field descriptions (continued)

Field	Description
	This bit indicates whether the SDHC supports high speed mode and the host system can supply a SD Clock frequency from 25 MHz to 50 MHz. 0b High speed not supported. 1b High speed supported.
20 ADMAS	ADMA Support This bit indicates whether the SDHC supports the ADMA feature. 0b Advanced DMA not supported. 1b Advanced DMA supported.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 MBL	Max Block Length This value indicates the maximum block size that the host driver can read and write to the buffer in the SDHC. The buffer shall transfer block size without wait cycles. 000b 512 bytes 001b 1024 bytes 010b 2048 bytes 011b 4096 bytes
15–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

50.4.18 Watermark Level register (SDHCx_WML)

Both write and read watermark levels (FIFO threshold) are configurable. There value can range from 1 to 128 words. Both write and read burst lengths are also configurable. There value can range from 1 to 31 words.

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WRBRSTLEN					WRWML								0			RDBRSTLEN					RDWML							
W																																
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

SDHCx_WML field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 WRBRSTLEN	Write Burst Length [1]

Table continues on the next page...

SDHCx_WML field descriptions (continued)

Field	Description
	The number of words the SDHC writes in a single burst. The write burst length must be less than or equal to the write watermark level, and all bursts within a watermark level transfer will be in back-to-back mode. On reset, this field will be 8. Writing 0 to this field will result in '01000' (i.e. it is not able to clear this field).
23–16 WRWML	Write Watermark Level The number of words used as the watermark level (FIFO threshold) in a DMA write operation. Also the number of words as a sequence of write bursts in back-to-back mode. The maximum legal value for the write watermark level is 128.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 RDBRSTLEN	Read Burst Length [2] The number of words the SDHC reads in a single burst. The read burst length must be less than or equal to the read watermark level, and all bursts within a watermark level transfer will be in back-to-back mode. On reset, this field will be 8. Writing 0 to this field will result in '01000' (i.e. it is not able to clear this field).
7–0 RDWML	Read Watermark Level The number of words used as the watermark level (FIFO threshold) in a DMA read operation. Also the number of words as a sequence of read bursts in back-to-back mode. The maximum legal value for the read water mark level is 128.

1. Due to system restriction, the actual burst length may not exceed 16.
2. Due to system restriction, the actual burst length may not exceed 16.

50.4.19 Force Event register (SDHCx_FEVT)

The Force Event (FEVT) register is not a physically implemented register. Rather, it is an address at which the Interrupt Status register can be written if the corresponding bit of the Interrupt Status Enable register is set. This register is a write only register and writing 0 to it has no effect. Writing 1 to this register actually sets the corresponding bit of Interrupt Status register. A read from this register always results in 0's. To change the corresponding status bits in the interrupt status register, make sure to set SYSCTL[IPGEN] so that bus clock is always active.

Forcing a card interrupt will generate a short pulse on the DAT[1] line, and the driver may treat this interrupt as a normal interrupt. The interrupt service routine may skip polling the card interrupt factor as the interrupt is selfcleared.

Address: Base address + 50h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			0				0		0	0	0	0	0	0	0
W	CINT	0		DMAE	0			AC12E	0	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R									0			0	0	0	0	0
W	0								CNIBAC12E	0		AC12IE	AC12EBE	AC12CE	AC12TOE	AC12NE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_FEVT field descriptions

Field	Description
31 CINT	Force Event Card Interrupt Writing 1 to this bit generates a short low-level pulse on the internal DAT[1] line, as if a self-clearing interrupt was received from the external card. If enabled, the CINT bit will be set and the interrupt service routine may treat this interrupt as a normal interrupt from the external card.
30–29 Reserved	This field is reserved.
28 DMAE	Force Event DMA Error Forces the DMAE bit of Interrupt Status Register to be set.
27–25 Reserved	This field is reserved.
24 AC12E	Force Event Auto Command 12 Error Forces IRQSTAT[AC12E] to be set.
23 Reserved	This field is reserved.
22 DEBE	Force Event Data End Bit Error Forces IRQSTAT[DEBE] to be set.
21 DCE	Force Event Data CRC Error

Table continues on the next page...

SDHCx_FEVT field descriptions (continued)

Field	Description
	Forces IRQSTAT[DCE] to be set.
20 DTOE	Force Event Data Time Out Error Forces IRQSTAT[DTOE] to be set.
19 CIE	Force Event Command Index Error Forces IRQSTAT[CCE] to be set.
18 CEBE	Force Event Command End Bit Error Forces IRQSTAT[CEBE] to be set.
17 CCE	Force Event Command CRC Error Forces IRQSTAT[CCE] to be set.
16 CTOE	Force Event Command Time Out Error Forces IRQSTAT[CTOE] to be set.
15–8 Reserved	This field is reserved.
7 CNIBAC12E	Force Event Command Not Executed By Auto Command 12 Error Forces AC12ERR[CNIBAC12E] to be set.
6–5 Reserved	This field is reserved.
4 AC12IE	Force Event Auto Command 12 Index Error Forces AC12ERR[AC12IE] to be set.
3 AC12EBE	Force Event Auto Command 12 End Bit Error Forces AC12ERR[AC12EBE] to be set.
2 AC12CE	Force Event Auto Command 12 CRC Error Forces AC12ERR[AC12CE] to be set.
1 AC12TOE	Force Event Auto Command 12 Time Out Error Forces AC12ERR[AC12TOE] to be set.
0 AC12NE	Force Event Auto Command 12 Not Executed Forces AC12ERR[AC12NE] to be set.

50.4.20 ADMA Error Status register (SDHCx_ADMAES)

When an ADMA error interrupt has occurred, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address register holds the address around the error descriptor.

For recovering from this error, the host driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address.
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address.
- ST_CADR: This state is never set because it only increments the descriptor pointer and doesn't generate an ADMA error.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address.

In case of a write operation, the host driver must use the ACMD22 to get the number of the written block, rather than using this information, because unwritten data may exist in the host controller.

The host controller generates the ADMA error interrupt when it detects invalid descriptor data (valid = 0) in the ST_FDS state. The host driver can distinguish this error by reading the valid bit of the error descriptor.

Table 50-63. ADMA Error State coding

D01-D00	ADMA Error State when error has occurred	Contents of ADMA System Address register
00	ST_STOP (Stop DMA)	Holds the address of the next executable descriptor command
01	ST_FDS (fetch descriptor)	Holds the valid descriptor address
10	ST_CADR (change address)	No ADMA error is generated
11	ST_TFR (Transfer Data)	Holds the address of the next executable descriptor command

Address: Base address + 54h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ADMADCE	ADMALME	ADMAES	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_ADMAES field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ADMADCE	ADMA Descriptor Error This error occurs when an invalid descriptor is fetched by ADMA. 0b No error. 1b Error.
2 ADMALME	ADMA Length Mismatch Error This error occurs in the following 2 cases: • While the block count enable is being set, the total data length specified by the descriptor table is different from that specified by the block count and block length. • Total data length can not be divided by the block length. 0b No error. 1b Error.
1–0 ADMAES	ADMA Error State (When ADMA Error Is Occurred.) Indicates the state of the ADMA when an error has occurred during an ADMA data transfer.

50.4.21 ADMA System Addressregister (SDHCx_ADSADDR)

This register contains the physical system memory address used for ADMA transfers.

Address: Base address + 58h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADSADDR																														0	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_ADSADDR field descriptions

Field	Description
31–2 ADSADDR	ADMA System Address Holds the word address of the executing command in the descriptor table. At the start of ADMA, the host driver shall set the start address of the Descriptor table. The ADMA engine increments this register address whenever fetching a descriptor command. When the ADMA is stopped at the block gap, this register indicates the address of the next executable descriptor command. When the ADMA error interrupt is generated, this register shall hold the valid descriptor address depending on the ADMA state. The lower 2 bits of this register is tied to ‘0’ so the ADMA address is always word-aligned. Because this register supports dynamic address reflecting, when TC bit is set, it automatically alters the value of internal address counter, so SW cannot change this register when TC bit is set.
1–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

50.4.22 Vendor Specific register (SDHCx_VENDOR)

This register contains the vendor-specific control/status register.

Address: Base address + C0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				0				INTSTVAL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														EXBLKNU	Reserved
W																Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SDHCx_VENDOR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 INTSTVAL	Internal State Value Internal state value, reflecting the corresponding state value selected by **Debug Select** field. This field is read-only and write to this field does not have effect.
15–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 EXBLKNU	Exact Block Number Block Read Enable For SDIO CMD53 This bit must be set before S/W issues CMD53 multi-block read with exact block number. This bit must not be set if the CMD53 multi-block read is not exact block number. 0 None exact block read. 1 Exact block read for SDIO CMD53.
0 Reserved	This field is reserved.

50.4.23 MMC Boot register (SDHCx_MMCBOOT)

This register contains the MMC fast boot control register.

Address: Base address + C4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	BOOTBLKCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								AUTOSABGEN	BOOTEN	BOOTMODE	BOOTACK	DTOCVACK			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDHCx_MMCBOOT field descriptions

Field	Description
31–16 BOOTBLKCNT	Defines the stop at block gap value of automatic mode. When received card block cnt is equal to BOOTBLKCNT and AUTOSABGEN is 1, then stop at block gap.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 AUTOSABGEN	When boot, enable auto stop at block gap function. This function will be triggered, and host will stop at block gap when received card block cnt is equal to BOOTBLKCNT.
6 BOOTEN	Boot Mode Enable 0 Fast boot disable. 1 Fast boot enable.
5 BOOTMODE	Boot Mode Select 0 Normal boot. 1 Alternative boot.
4 BOOTACK	Boot Ack Mode Select 0 No ack. 1 Ack.
3–0 DTOCVACK	Boot ACK Time Out Counter Value 0000b SDCLK x 2^8 0001b SDCLK x 2^9 0010b SDCLK x 2^10 0011b SDCLK x 2^11 0100b SDCLK x 2^12

Table continues on the next page...

SDHCx_MMCBOOT field descriptions (continued)

Field	Description
	0101b SDCLK x 2^13 0110b SDCLK x 2^14 0111b SDCLK x 2^15 ... 1110b SDCLK x 2^22 1111b Reserved

50.4.24 Host Controller Version (SDHCx_HOSTVER)

This register contains the vendor host controller version information. All bits are read only and will read the same as the power-reset value.

Address: Base address + FCh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																VVN								SVN							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1

SDHCx_HOSTVER field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 VVN	Vendor Version Number These status bits are reserved for the vendor version number. The host driver shall not use this status. 00h Freescale SDHC version 1.0 10h Freescale SDHC version 2.0 11h Freescale SDHC version 2.1 12h Freescale SDHC version 2.2 All others Reserved
7–0 SVN	Specification Version Number These status bits indicate the host controller specification version. 01h SD host specification version 2.0, supports test event register and ADMA. All others Reserved

50.5 Functional description

The following sections provide a brief functional description of the major system blocks, including the data buffer, DMA crossbar switch interface, dual-port memory wrapper, data/command controller, clock & reset manager, and clock generator.

50.5.1 Data buffer

The SDHC uses one configurable data buffer, so that data can be transferred between the system bus and the SD card, with an optimized manner to maximize throughput between the two clock domains (that is, the IP peripheral clock, and the master clock). The following diagram illustrates the buffer scheme. The buffer is used as temporary storage for data being transferred between the host system and the card. The watermark levels for read and write are both configurable, and can be any number from 1 to 128 words. The burst lengths for read and write are also configurable, and can be any number from 1 to 31 words.

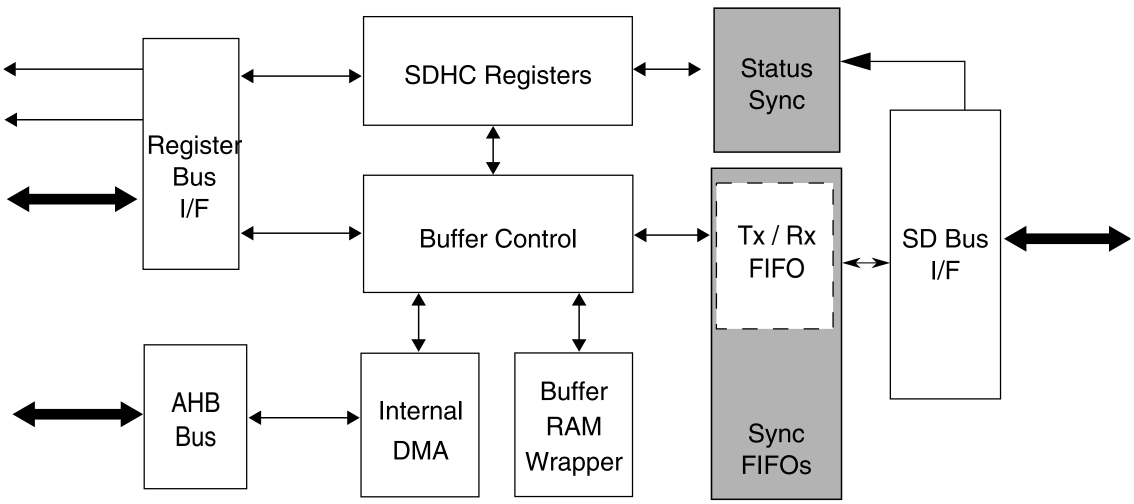


Figure 50-75. SDHC buffer scheme

There are 3 transfer modes to access the data buffer:

- CPU Polling mode:

    - For a host read operation, when the number of words received in the buffer meets or exceeds the RDWML watermark value, then by polling IRQSTAT[BRR] the host driver can read the DATPORT register to fetch the amount of words set in the WML register from the buffer. The write operation is similar.
- Internal DMA mode (includes simple and advanced DMA access's):
    - The internal DMA access, either by simple or advanced DMA, is over the crossbar switch bus. For internal DMA access mode, the external DMA request will never be sent out.

For a read operation, when there are more words in the buffer than the amount set in WML, the internal DMA starts fetching data over the crossbar switch bus. Except INCR4 and INCR8, the burst type is always INCR mode and the burst length depends on the shortest of following factors:

1. Burst length configured in the burst length field of WML
2. Watermark level boundary
3. Block size boundary
4. Data boundary configured in the current descriptor if the ADMA is active
5. 1 KB address boundary

Write operation is similar.

Sequential and contiguous access is necessary to ensure the pointer address value is correct. Random or skipped access is not possible. The byte order, by reset, is little endian mode. The actually byte order is swapped inside the buffer, according to the endian mode configured by software, as illustrated in the following diagrams. For a host write operation, byte order is swapped after data is fetched from the buffer and ready to send to the SD bus. For a host read operation, byte order is swapped before the data is stored into the buffer.

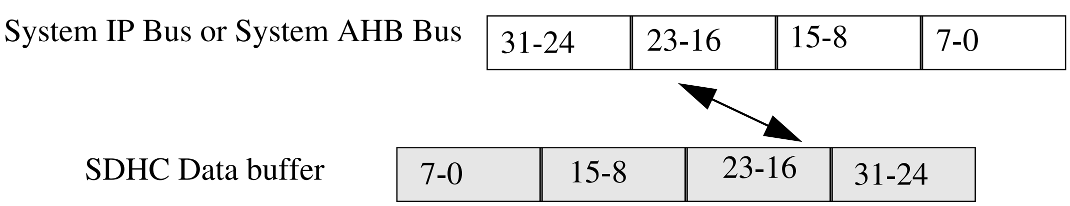


Figure 50-76. Data swap between system bus and SDHC data buffer in byte little endian mode

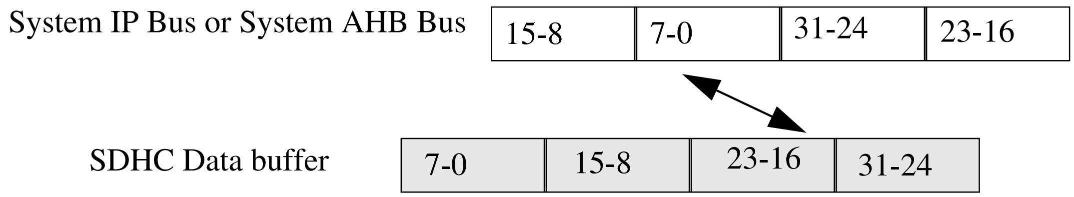


Figure 50-77. Data swap between system bus and SDHC data buffer in half word big endian mode

50.5.1.1 Write operation sequence

There are three ways to write data into the buffer when the user transfers data to the card:

1. Using external DMA through the SDHC DMA request signal.
2. Processor core polling through IRQSTAT[BWR] (interrupt or polling).
3. Using the internal DMA.

When the internal DMA is not used, that is, the XFERTYP[DMAEN] bit is not set when the command is sent, the SDHC asserts a DMA request when the amount of buffer space exceeds the value set in WML, and is ready for receiving new data. At the same time, the SDHC would set IRQSTAT[BWR]. The buffer write ready interrupt will be generated if it is enabled by software.

When internal DMA is used, the SDHC will not inform the system before all the required number of bytes are transferred if no error was encountered. When an error occurs during the data transfer, the SDHC will abort the data transfer and abandon the current block. The host driver must read the contents of the DSADDR to get the starting address of the abandoned data block. If the current data transfer is in multiblock mode, the SDHC will not automatically send CMD12, even though XFERTYP[AC12EN] is set. The host driver shall send CMD12 in this scenario and restart the write operation from that address. It is recommended that a software reset for data be applied before the transfer is restarted after error recovery.

The SDHC will not start data transmission until the number of words set in WML can be held in the buffer. If the buffer is empty and the host system does not write data in time, the SDHC will stop the SD_CLK to avoid the data buffer underrun situation.

50.5.1.2 Read operation sequence

There are three ways to read data from the buffer when the user transfers data to the card:

1.

2. Processor core polling through IRQSTAT[BRR] (interrupt or polling)
3. Using the internal DMA

When internal DMA is not used, that is, XFERTYP[DMAEN] is not set when the command is sent, the SDHC asserts a DMA request when the amount of data exceeds the value set in WML, that is available and ready for system fetching data. At the same time, the SDHC would set IRQSTAT[BRR]. The buffer read ready interrupt will be generated if it is enabled by software.

When internal DMA is used, the SDHC will not inform the system before all the required number of bytes are transferred if no error was encountered. When an error occurs during the data transfer, the SDHC will abort the data transfer and abandon the current block. The host driver must read the content of the DMA system address register to get the starting address of the abandoned data block. If the current data transfer is in multiblock mode, the SDHC will not automatically send CMD12, even though XFERTYP[AC12EN] is set. The host driver shall send CMD12 in this scenario and restart the read operation from that address. It is recommended that a software reset for data be applied before the transfer is restarted after error recovery.

For any write transfer mode, the SDHC will not start data transmission until the number of words set in WML are in the buffer. If the buffer is full and the Host System does not read data in time, the SDHC will stop the SDHC_DCLK to avoid the data buffer overrun situation.

50.5.1.3 Data buffer and block size

The user needs to know the buffer size for the buffer operation during a data transfer to use it in the most optimized way. In the SDHC, the only data buffer can hold up to 128 words (32-bit), and the watermark levels for write and read can be configured respectively. For both read and write, the watermark level can be from 1 word to the maximum of 128 words. For both read and write, the burst length can be from 1 word to the maximum of 31 words. The host driver may configure the value according to the system situation and requirement.

During a multiblock data transfer, the block length may be set to any value between 1 and 4096 bytes inclusive, which satisfies the requirements of the external card. The only restriction is from the external card. It might not support that large of a block or it doesn't support a partial block access, which is not the integer times of 512 bytes.

For block size not times of 4, that is, not word aligned, SDHC requires stuff bytes at the end of each block, because the SDHC treats each block individually. For example, the block size is 7 bytes, there are 12 blocks to write, the system side must write 2 times for

each block, and for each block, the ending byte would be abandoned by the SDHC because it sends only 7 bytes to the card and picks data from the following system write, so there would be 24 beats of write access in total.

50.5.1.4 Dividing large data transfer

This SDIO command CMD53 definition, limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in block size units. If the total data length can't be divided evenly into a multiple of the block size, then there are two ways to transfer the data which depend on the function and the card design. Option 1 is for the host driver to split the transaction. The remainder of the block size data is then transferred by using a single block command at the end. Option 2 is to add dummy data in the last block to fill the block size. For option 2, the card must manage the removal of the dummy data.

The following diagram illustrates the dividing of large data transfers. Assuming a kind of WLAN SDIO card supports block size only up to 64 bytes. Although the SDHC supports a block size of up to 4096 bytes, the SDIO can only accept a block size less than 64 bytes, so the data must be divided. See the example below.

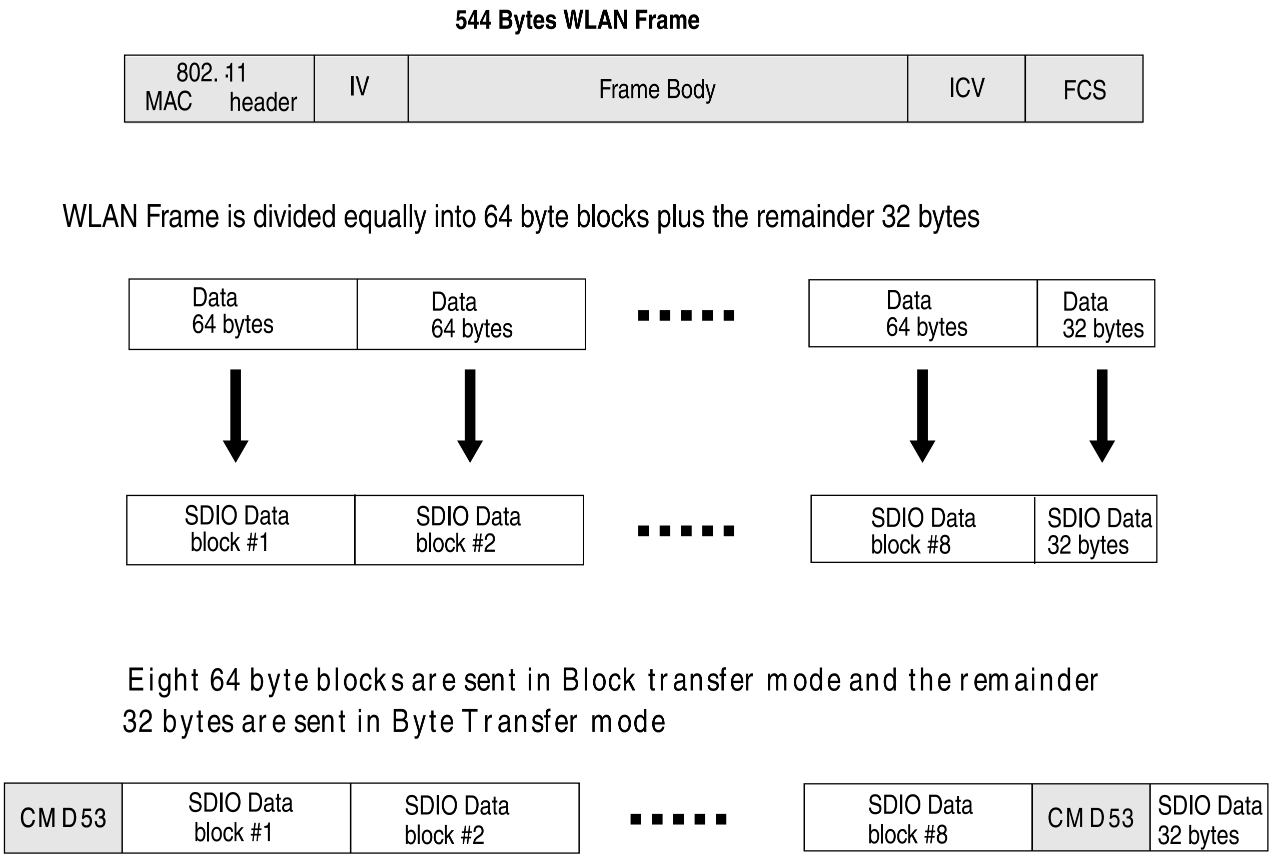


Figure 50-78. Example for dividing large data transfers

50.5.1.5 External DMA request

When the internal DMA is not in use, and external DMA request is enabled, the data buffer will generate a DMA request to the system. During a write operation, when the number of WRWML words can be held in the buffer free space, a DMA request is sent , informing the host system of a DMA write. IRQSTAT[BWR] is also set, as long as IRQSTATEN[BWRSEN] is set. The DMA request is immediately deasserted when an access to the DATPORT register is made. If the buffer's free space still meets the watermark condition, the DMA request is asserted again after a cycle.

On read operation, when the number of RDWML words are already in the buffer, a DMA request is sent, informing the host system for a DMA read. IRQSTAT[BRR] is also set, as long as IRQSTATEN[BRRSEN] is set. The DMA request is immediately deasserted when an access to the DATPORT register is made. If the buffer's data still meets the watermark condition, the DMA request is asserted again after a cycle.

Because the DMA burst length can't change during a data transfer for an external DMA transfer, the watermark level (read or write) must be a divisor of the block size. If it is not, transferring of the block may cause buffer underrun for read operation or overrun for

write operation. For example, if the block size is 512 bytes, the watermark level of read (or write) must be a power of 2 between 1 and 128. For processor core polling access, as the last access in the block transfer can be controlled by software, there is no such issue. The watermark level can be any value, even larger than the block size but not greater than 128 words. This is because the actual number of bytes transferred by the software can be controlled and does not exceed the block size in each transfer.

The SDHC also supports non-word aligned block size, as long as the card supports that block size. In this case, the watermark level must be set as the number of words. For example, if the block size is 31 bytes, the watermark level can be set to any number of word. For this case, the BLKATTR[BLKSIZE] bits shall be set as 1fh. For the CPU polling access, the burst length can be 1 to 128 words, without restriction. This is because the software will transfer 8 words, and the SDHC will also set the IRQSTAT[BWR] or IRQSTAT[BRR] bits when the remaining data does not violate data buffer. See DMA burst length for more details about the dynamic watermark level of the data buffer. For the above example, even though 8 words are transferred via the DATPORT register, the SDHC will transfer only 31 bytes over the SD bus, as required by the BLKATTR[BLKSIZE] bits. In this data transfer, with non-word aligned block size, the endian mode must be set cautiously, or invalid data will be transferred to/from the card.

50.5.2 DMA crossbar switch interface

The internal DMA implements a DMA engine and the crossbar switch master. When the internal DMA is enabled (XFERTYP[DMAEN] is set), the interrupt status bits are set if they are enabled. To avoid setting them, clear IRQSTATEN[BWRSEN, BRRSEN]. The following diagram illustrates the DMA crossbar switch interface block.

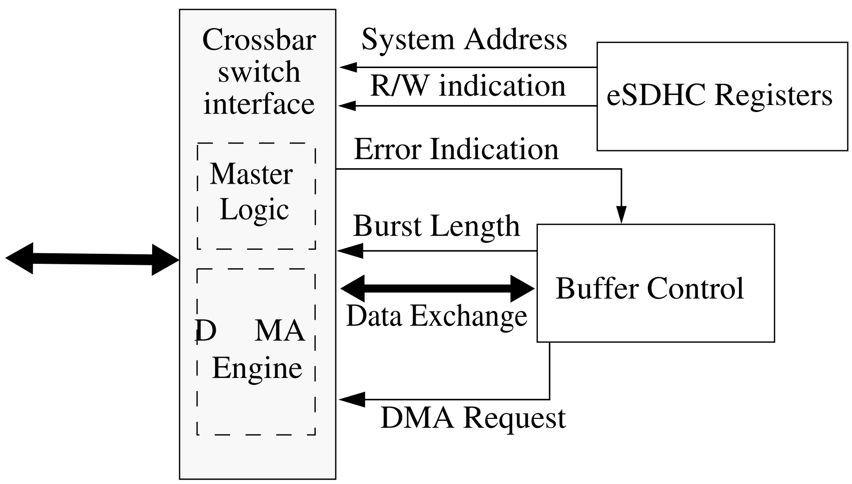


Figure 50-79. DMA crossbar switch interface block

50.5.2.1 Internal DMA request

If the watermark level requirement is met in data transfer, and the internal DMA is enabled, the data buffer block sends a DMA request to the crossbar switch interface. Meanwhile, the external DMA request signal is disabled. The delay in response from the internal DMA engine depends on the system bus loading and the priority assigned to the SDHC. The DMA engine does not respond to the request during its burst transfer, but is ready to serve as soon as the burst is over. The data buffer de-asserts the request once an access to the buffer is made. Upon access to the buffer by internal DMA, the data buffer updates its internal buffer pointer, and when the watermark level is satisfied, another DMA request is sent.

The data transfer is in the block unit, and the subsequent watermark level is always set as the remaining number of words. For instance, for a multiblock data read with each block size of 31 bytes, and the burst length set to 6 words. After the first burst transfer, if there are more than 2 words in the buffer, which might contain some data of the next block, another DMA request read is sent. This is because the remaining number of words to send for the current block is (31 - 6 * 4) / 4 = 2. The SDHC will read 2 words out of the buffer, with 7 valid bytes and 1 stuff byte.

50.5.2.2 DMA burst length

Just like a CPU polling access, the DMA burst length for the internal DMA engine can be from 1 to 16 words. The actual burst length for the DMA depends on the lesser of the configured burst length or the remaining words of the current block. Take the example in Internal DMA request again. The following burst length after 6 words are read will be 2 words, and the next burst length will be 6 words again. This is because the next block starts, which is 31 bytes, more than 6 words. The host driver writer may take this variable burst length into account. It is also acceptable to configure the burst length as the divisor of the block size, so that each time the burst length will be the same.

50.5.2.3 Crossbar switch master interface

It is possible that the internal DMA engine could fail during the data transfer. When this error occurs, the DMA engine stops the transfer and goes to the idle state as well as the internal data buffer stops accepting incoming data. IRQSTAT[DMAE] is set to inform the driver.

After the DMAE interrupt is received, the software shall send a CMD12 to abort the current transfer and read DSADDR[DSADDR] to get the starting address of the corrupted block. After the DMA error is fixed, the software must apply a data reset and restart the transfer from this address to recover the corrupted block.

50.5.2.4 ADMA engine

In the SDHC standard, the new DMA transfer algorithm called the advanced DMA (ADMA) is defined. For simple DMA, once the page boundary is reached, a DMA interrupt will be generated and the new system address shall be programmed by the host driver. The ADMA defines the programmable descriptor table in the system memory. The host driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. It reduces the frequency of interrupts to the host system. Therefore, higher speed DMA transfers could be realized because the host MCU intervention would not be needed during long DMA-based data transfers.

There are two types of ADMA: ADMA1 and ADMA2 in host controller. ADMA1 can support data transfer of 4 KB aligned data in system memory. ADMA2 improves the restriction so that data of any location and any size can be transferred in system memory. Their formats of descriptor table are different.

ADMA can recognize all kinds of descriptors defined in the SDHC standard, and if end flag is detected in the descriptor, ADMA will stop after this descriptor is processed.

50.5.2.4.1 ADMA concept and descriptor format

For ADMA1, including the following descriptors:

- Valid/Invalid descriptor
- Nop descriptor
- Set data length descriptor
- Set data address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

For ADMA2, including the following descriptors:

- Valid/Invalid descriptor
- Nop descriptor

- Rsv descriptor
- Set data length and address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the Valid flag of descriptor is 0, it will ignore the high 32-bit. The address field shall be set on word aligned (lower 2-bit is always set to 0). Data length is in byte unit.

ADMA will start read/write operation after it reaches the tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last set type descriptor before tran type descriptor. Every tran type will trigger a transfer, and the transfer data length is extracted from the most recent set type descriptor. If there is no set type descriptor after the previous trans descriptor, the data length will be the value for previous transfer, or 0 if no set descriptor is ever met.

For ADMA2, tran type descriptor contains both the data length and transfer data address, so only a tran type descriptor can start a data transfer.

Table 50-101. Format of the ADMA1 descriptor table

Address/page field		**Address/page field**		**Attribute field**					
31	12	11	6	5	4	3	2	1	0
Address or data length		000000		Act2	Act1	0	Int	End	Valid

Act2	**Act1**	**Symbol**	**Comment**	**31-28**	**27-12**
0	0	nop	No operation	Don't care	
0	1	set	Set data length	0000	Data length
1	0	tran	Transfer data	Data address	
1	1	link	Link descriptor	Descriptor address	

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA interrupt when this descriptor is processed.

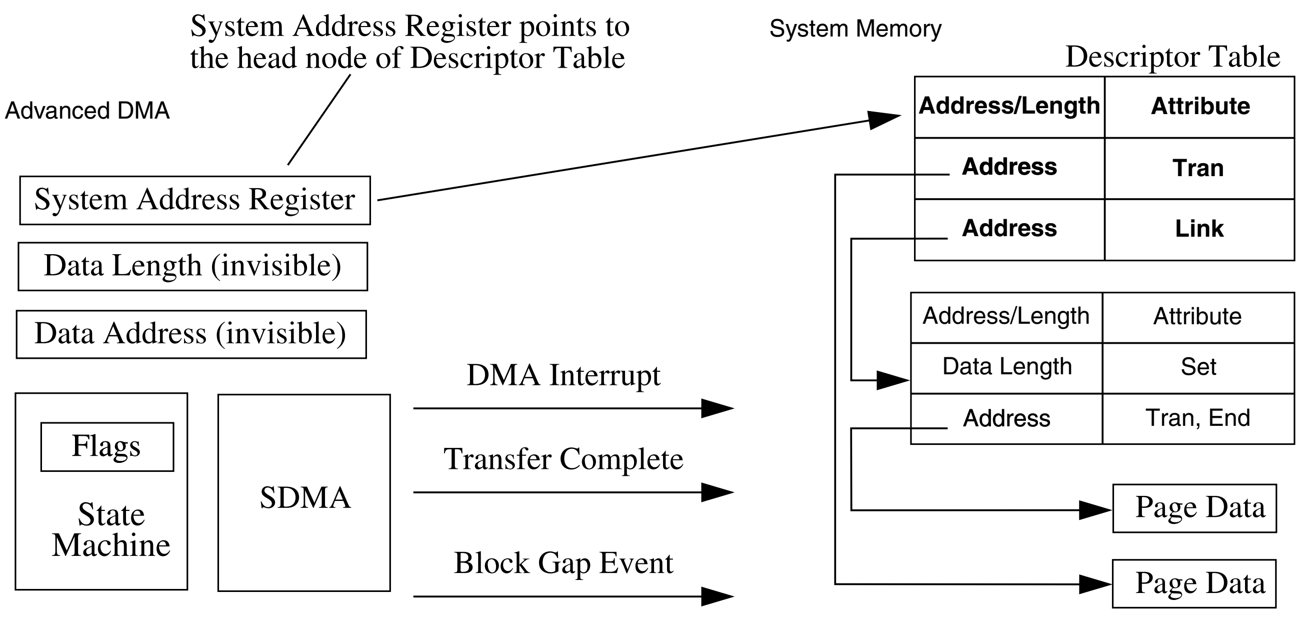


Figure 50-80. Concept and access method of ADMA1 descriptor table

Table 50-102. Format of the ADMA2 descriptor table

Address field		**Length**		**Reserved**			**Attribute field**				
63	32	31	16	15	06	05	04	03	02	01	00
32-bit address		16-bit length		0000000000		Act2	Act1	0	Int	End	Valid

Act2	**Act1**	**Symbol**	**Comment**	**Operation**
0	0	nop	No operation	Don't care
0	1	rsv	Reserved	Same as nop. Read this line and go to next one
1	0	tran	Transfer data	Transfer data with address and length set in this descriptor line
1	1	link	Link descriptor	Link to another descriptor

Valid	Valid = 1 indicates this line of descriptor is effective. If valid = 0 generate ADMA error interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA interrupt when this descriptor is done.

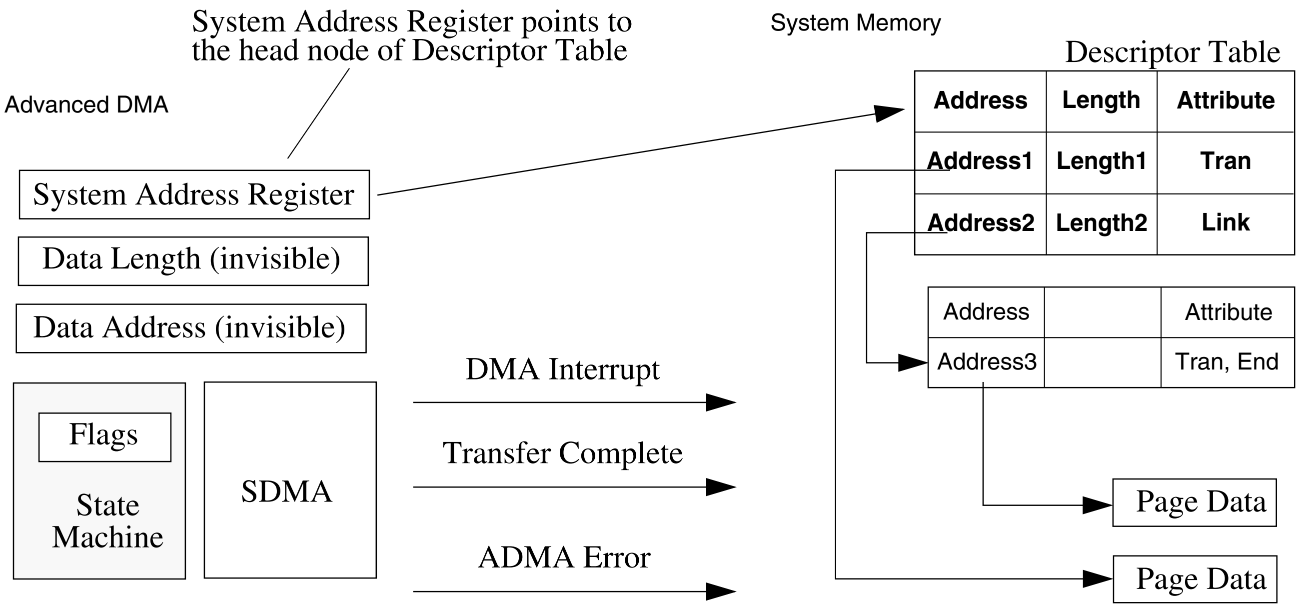


Figure 50-81. Concept and access method of ADMA2 descriptor table

50.5.2.4.2 ADMA interrupt

If the interrupt flag of descriptor is set, ADMA will generate an interrupt according to different type descriptor:

For ADMA1:

- Set type descriptor: interrupt is generated when data length is set.
- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type descriptor: interrupt is generated when new descriptor address is set.
- Nop type descriptor: interrupt is generated just after this descriptor is fetched.

For ADMA2:

- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type descriptor: interrupt is generated when new descriptor address is set.
- Nop/Rsv type descriptor: interrupt is generated just after fetch this descriptor.

50.5.2.4.3 ADMA error

The ADMA stops whenever any error is encountered. These errors include:

- Fetching descriptor error

- Transfer error
- Data length mismatch error

ADMA descriptor error will be generated when it fails to detect the valid flag in the descriptor. If ADMA descriptor error occurs, the interrupt is not generated even if the interrupt flag of this descriptor is set.

When XFERTYP[BCEN] is set, data length set in buffer must be equal to the whole data length set in descriptor nodes, otherwise data length mismatch error will be generated.

If XFERTYP[BCEN] is not set, the whole data length set in descriptor must be times of block length, otherwise, when all data set in the descriptor nodes are done not at block boundary, the data mismatch error will occur.

50.5.3 SD protocol unit

The SD protocol unit deals with all SD protocol affairs.

The SD protocol unit performs the following functions:

- Acts as the bridge between the internal buffer and the SD bus
- Sends the command data as well as its argument serially
- Stores the serial response bit stream into the corresponding registers
- Detects the bus state on the DAT[0] line
- Monitors the interrupt from the SDIO card
- Asserts the read wait signal
- Gates off the SD clock when buffer is announcing danger status
- Detects the write protect state

The SD protocol unit consists of four submodules:

- SD transceiver
- SD clock and monitor
- Command agent
- Data agent

50.5.3.1 SD transceiver

In the SD protocol unit, the transceiver is the main control module. It consists of a FSM and control module, from which the control signals for all other three modules are generated.

50.5.3.2 SD clock & monitor

This module monitors the signal level on all 8 data lines, the command lines, and directly routes the level values into the register bank. The driver can use this for debug purposes.

The module also detects the card detection (CD) line as well as the DAT[3] line. The transceiver reports the card insertion state according to the CD state, or the signal level on the DAT[3] line, when PROCTL[D3CD] is set.

The module detects the write protect (WP) line. With the information of the WP state, the register bank will ignore the command, accompanied by a write operation, when the WP switch is on.

If the internal data buffer is in danger, and the SD clock must be gated off to avoid buffer over/under-run, this module will assert the gate of the output SD clock to shut the clock off. After the buffer danger has recovered, and when the system access of the buffer catches up, the clock gate of this module will open and the SD clock will be active again.

This module also drives the SDHC_LCTL output signal when PROCTL[LCTL] is set by the driver.

50.5.3.3 Command agent

The command agent deals with the transactions on the CMD line. The following diagram illustrates the structure for the command CRC Shift Register.

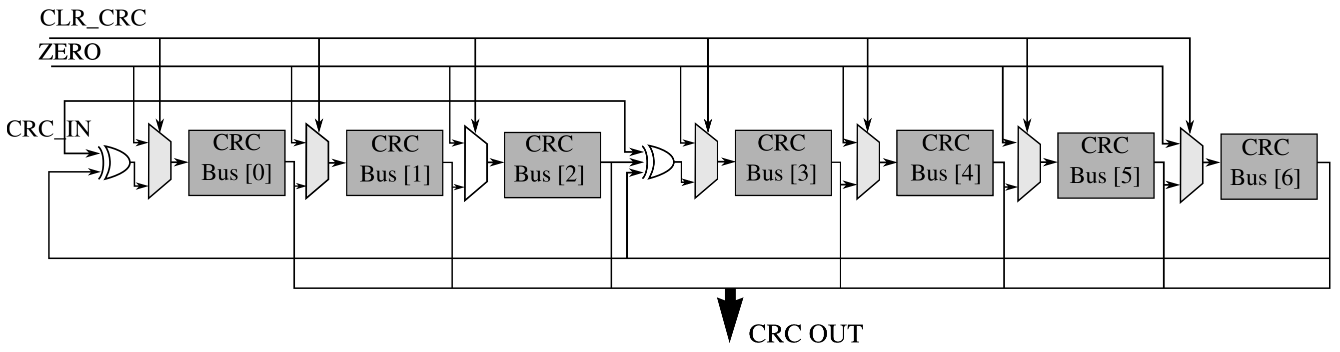


Figure 50-82. Command CRC Shift Register

The CRC polynomials for the CMD line are as follows:

Generator polynomial: $G(x) = x^7 + x^3 + 1$
$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} +...+ (\text{last bit}) * x^0$
$CRC[6:0] = \text{Remainder } [(M(x) * x^7) / G(x)]$

50.5.3.4 Data agent

The data agent deals with the transactions on the eight data lines. Moreover, this module also detects the busy state on the DAT[0] line, and generates the read wait state by the request from the transceiver. The CRC polynomials for the DAT are as follows:

Generator polynomial: $G(x) = x^{16} + x^{12} + x^5 +1$
$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} +...+ (\text{last bit}) * x^0$
$CRC[15:0] = \text{Remainder } [(M(x) * x^{16}) / G(x)]$

50.5.4 Clock and reset manager

This module controls all the reset signals within the SDHC.

There are four kinds of reset signals within the SDHC:

- Hardware reset
- Software reset for all
- Software reset for the data part
- Software reset for the command part

All these signals are fed into this module and stable signals are generated inside the module to reset all other modules. The module also gates off all the inside signals.

There are three clocks inside the SDHC:

- Bus clock
- SDHC clock
- System clock

The module monitors the activities of all other modules, supplies the clocks for them, and when enabled, automatically gates off the corresponding clocks.

50.5.5 Clock generator

The clock generator generates the SDHC_CLK by peripheral source clock in two stages. The following diagram illustrates the structure of the divider. The term "base" represents the frequency of peripheral source clock.

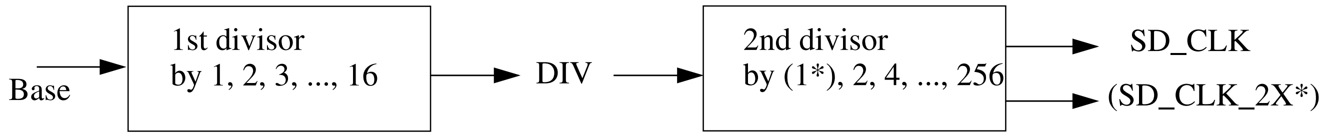


Figure 50-83. Two stages of the clock divider

The first stage outputs an intermediate clock (DIV), which can be base, base/2, base/3, ..., or base/16.

The second stage is a prescaler, and outputs the actual clock (SDHC_CLK). This clock is the driving clock for all submodules of the SD protocol unit, and the sync FIFOs to synchronize with the data rate from the internal data buffer. The frequency of the clock output from this stage, can be DIV, DIV/2, DIV/4,..., or DIV/256. Thus, the highest frequency of the SDHC_CLK is base, and the next highest is base/2, while the lowest frequency is base/4096. If the base clock is of equal duty ratio (usually true), the duty cycle of SDHC_CLK is also 50%, even when the compound divisor is an odd value.

50.5.6 SDIO card interrupt

This section discusses SDIO interrupt handling.

50.5.6.1 Interrupts in 1-bit mode

In this case, the DAT[1] pin is dedicated to providing the interrupt function. An interrupt is asserted by pulling the DAT[1] low from the SDIO card, until the interrupt service is finished to clear the interrupt.

50.5.6.2 Interrupt in 4-bit mode

Because the interrupt and data line 1 share Pin 8 in 4-bit mode, an interrupt will be sent by the card and recognized by the host only during a specific time. This is known as the interrupt period. The SDHC will only sample the level on pin 8 during the interrupt period. At all other times, the host will ignore the level on pin 8, and treat it as the data signal. The definition of the interrupt period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the interrupt period becomes active two clock cycles after the completion of a data packet. This interrupt period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in 4-bit mode, there is only a limited period of time that the interrupt period can be active due to the limited period of data line availability between the multiple blocks of data. This requires a more strict definition of the interrupt period. In this case, the interrupt period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the SDHC_D1 line will be held low for one clock cycle with the last clock cycle pulling SDHC_D1 high. On completion of the Interrupt Period, the card releases the SDHC_D1 line into the high Z state. The SDHC samples the SDHC_D1] during the interrupt period when PROCTL[IABG] is set.

See the SDIO Card Specification v1.10f for further information about the SDIO card interrupt.

50.5.6.3 Card interrupt handling

When IRQSIGEN[CINTIEN] is set to 0, the SDHC clears the interrupt request to the host system. The host driver must clear this bit before servicing the SDIO Interrupt and must set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

The SDIO status bit is cleared by resetting the SDIO interrupt. Writing to this bit would have no effects. In 1-bit mode, the SDHC will detect the SDIO interrupt with or without the SD clock (to support wakeup). In 4-bit mode, the interrupt signal is sampled during the interrupt period, so there are some sample delays between the interrupt signal from the SDIO card and the interrupt to the host system interrupt controller. When the SDIO status has been set, and the host driver needs to service this interrupt, so the SDIO bit in the interrupt control register of the SDIO card will be cleared. This is required to clear the

SDIO interrupt status latched in the SDHC and to stop driving the interrupt signal to the system interrupt controller. The host driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO Interrupt Enable bit is set to 1, and the SDHC starts sampling the interrupt signal again.

The following diagram illustrates the SDIO card interrupt scheme and for the sequences of software and hardware events that take place during a card interrupt handling procedure.

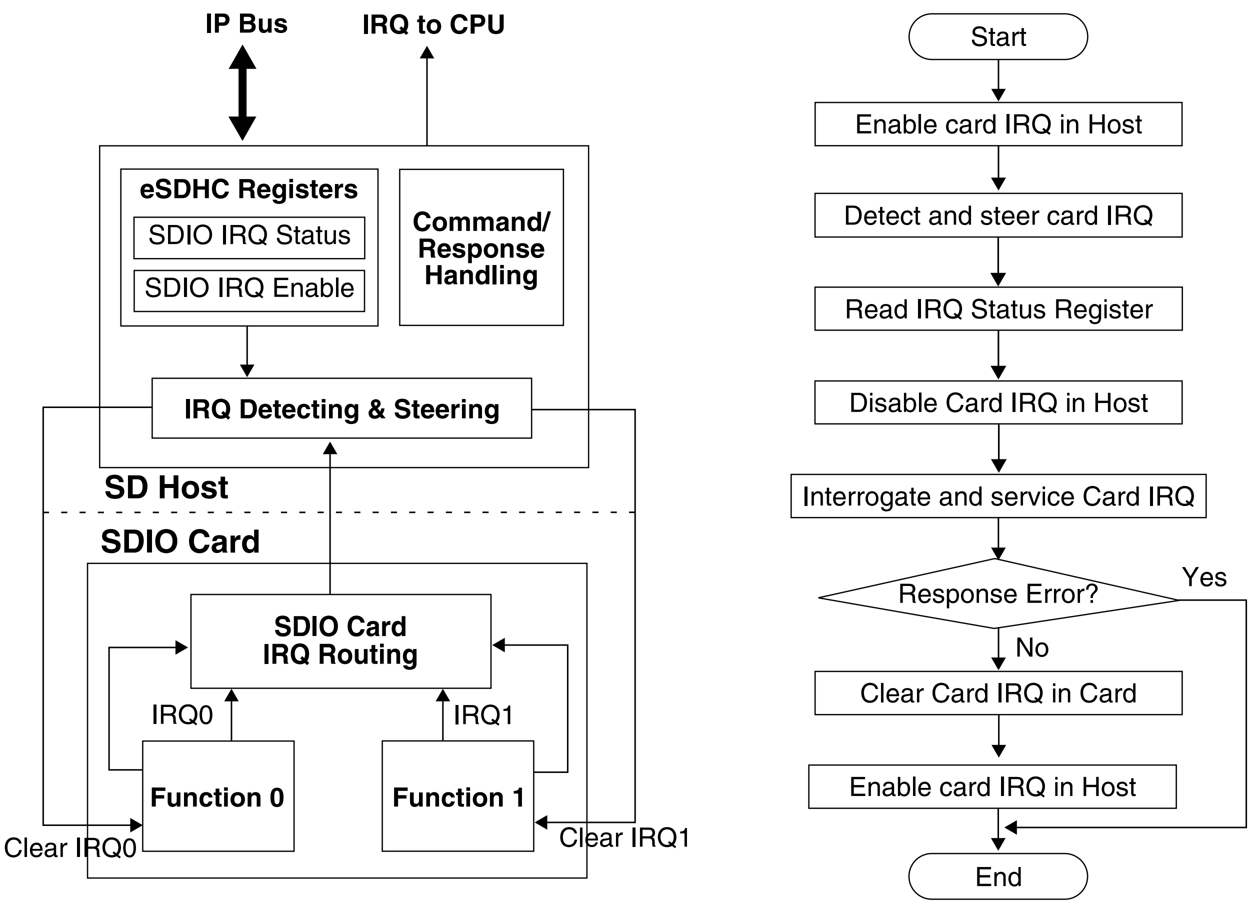


Figure 50-84. Card interrupt scheme and card interrupt detection and handling procedure

50.5.7 Card insertion and removal detection

The SDHC uses either the DAT[3] pin or the CD pin to detect card insertion or removal. When there is no card on the MMC/SD bus, the DAT[3] will be pulled to a low voltage level by default. When any card is inserted to or removed from the socket, the SDHC detects the logic value changes on the DAT[3] pin and generates an interrupt. When the DAT[3] pin is not used for card detection (for example, it is implemented in GPIO), the CD pin must be connected for card detection. Whether DAT[3] is configured for card

detection or not, the CD pin is always a reference for card detection. Whether the DAT[3] pin or the CD pin is used to detect card insertion, the SDHC will send an interrupt (if enabled) to inform the Host system that a card is inserted.

50.5.8 Power management and wakeup events

When there is no operation between the SDHC and the card through the SD bus, the user can completely disable the bus clock and the SDHC clock in the chip-level clock control module to save power. When the user needs to use the SDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to the SDHC are disabled, for instance, when the system is in low-power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The SDHC can generate these interrupt even when there are no clocks enabled.

The three interrupts which can be used as wakeup events are:

- Card removal interrupt
- Card insertion interrupt
- Interrupt from SDIO card

The SDHC offers a power management feature. By clearing the clock enabled bits in the system control register, the clocks are gated in the low position to the SDHC. For maximum power saving, the user can disable all the clocks to the SDHC when there is no operation in progress.

These three wakeup events, or wakeup interrupts, can also be used to wake up the system from low-power modes.

Note

To make the interrupt a wakeup event, when all the clocks to the SDHC are disabled or when the whole system is in low-power mode, the corresponding wakeup enabled bit needs to be set. See Protocol Control register for more information.

50.5.8.1 Setting wakeup events

For the SDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the CPU enters Sleep mode. Before the software disables the host clock, it must ensure that all of the following conditions have been met:

- No read or write transfer is active
- Data and command lines are not active
- No interrupts are pending
- Internal data buffer is empty

50.5.9 MMC fast boot

In Embedded MultiMediaCard(eMMC4.3) spec, add fast boot feature needs hardware support.

In Boot Operation mode, the master (multimediacard host) can read boot data from the slave (MMC device) by keeping CMD line low after power-on, or sending CMD0 with argument + 0xFFFFFFFA (optional for slave), before issuing CMD1.

There are two types of Fast Boot mode, boot operation, and Alternative boot operation in eMMC4.3 spec. Each type also has with acknowledge and without acknowledge modes.

Note

For the eMMC4.3 card setting, please see the eMMC4.3 specification.

50.5.9.1 Boot operation

Note

In this block guide, this fast boot is called Normal Fast Boot mode.

If the CMD line is held low for 74 clock cycles and more after power-up before the first command is issued, the slave recognizes that Boot mode is being initiated and starts preparing boot data internally.

Within 1 second after the CMD line goes low, the slave starts to send the first boot data to the master on the DAT line(s). The master must keep the CMD line low to read all of the boot data.

If boot acknowledge is enabled, the slave has to send acknowledge pattern 010 to the master within 50 ms after the CMD line goes low. If boot acknowledge is disabled, the slave will not send out acknowledge pattern '010'.

The master can terminate Boot mode with the CMD line high.

Boot operation will be terminated when all contents of the enabled boot data are sent to the master. After boot operation is executed, the slave shall be ready for CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.

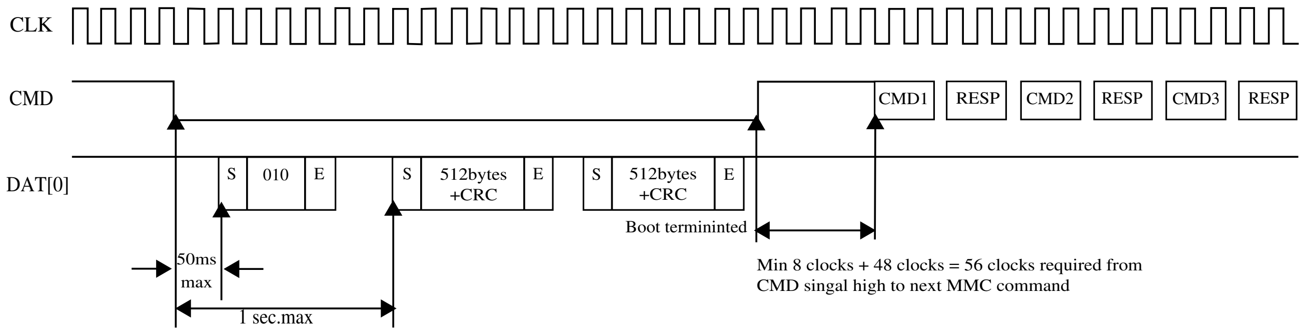


Figure 50-85. Multimediacard state diagram in Normal Boot mode

50.5.9.2 Alternative boot operation

This boot function is optional for the device. If bit 0 in the extended CSD byte[228] is set to 1, the device supports the alternative boot operation.

After power-up, if the host issues CMD0 with the argument of 0xFFFFFFFA after 74 clock cycles, before CMD1 is issued, or the CMD line goes low, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within 1 second after CMD0 with the argument of 0xFFFFFFFA is issued, the slave starts to send the first boot data to the master on the DAT line(s).

If boot acknowledge is enabled, the slave has to send the acknowledge pattern '010' to the master within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received. If boot acknowledge is disabled, theslave will not send out acknowledge pattern '010'.

The master can terminate boot mode by issuing CMD0 (Reset).

Boot operation will be terminated when all contents of the enabled boot data are sent to the master. After boot operation is executed, the slave shall be ready for CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.

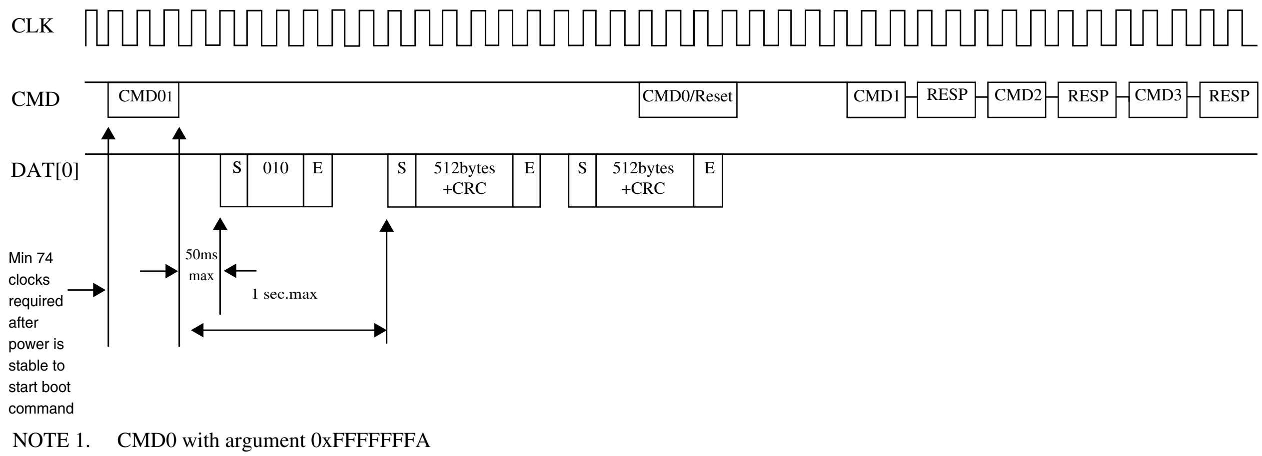


Figure 50-86. MultiMediaCard state diagram in Alternative Boot mode

50.6 Initialization/application of SDHC

All communication between system and cards are controlled by the host. The host sends commands of two types:

- Broadcast
- Addressed point-to-point

Broadcast commands are intended for all cards, such as GO_IDLE_STATE, SEND_OP_COND, ALL_SEND_CID, and so on. In Broadcast mode, all cards are in the Open-Drain mode to avoid bus contention. See Commands for MMC/SD/SDIO for the commands of bc and bcr categories.

After the broadcast command CMD3 is issued, the cards enter Standby mode. Addressed type commands are used from this point. In this mode, the CMD/DAT I/O pads will turn to Push-Pull mode, to have the driving capability for maximum frequency operation. See Commands for MMC/SD/SDIO for the commands of ac and adtc categories.

50.6.1 Command send and response receive basic operation

Assuming the data type WORD is an unsigned 32-bit integer, the following flow is a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)
{
WORD wCmd; // 32-bit integer to make up the data to write into Transfer Type register, it is
recommended to implement in a bit-field manner
wCmd = (<cmd_index> & 0x3f) >> 24; // set the first 8 bits as '00'+<cmd_index>
set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL accorind to the command index;
if (internal DMA is used) wCmd |= 0x1;
```

```
if (multi-block transfer) {
 set MSBSEL bit;
 if (finite block number) {
  set BCEN bit;
  if (auto12 command is to use) set AC12EN bit;
 }
}
write_reg(CMDARG, <cmd_arg>); // configure the command argument
write_reg(XFERTYP, wCmd); // set Transfer Type register as wCmd value to issue the command
}
wait_for_response(cmd_index)
{
while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set
read IRQ Status register and check if any error bits about Command are set
if (any error bits are set) report error;
write 1 to clear CC bit and all Command Error bits;
}
```

For the sake of simplicity, the function wait_for_response is implemented here by means of polling. For an effective and formal way, the response is usually checked after the command complete interrupt is received. By doing this, make sure the corresponding interrupt status bits are enabled.

In some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and go to the standby state, no response to the host when CMD2 is sent. The host driver shall deal with 'fake' errors like this with caution.

50.6.2 Card Identification mode

When a card is inserted to the socket or the card was reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the relative card address (RCA) or to set the RCA for the MMC cards.

50.6.2.1 Card detect

The following diagram illustrates the detection of MMC, SD, and SDIO cards using the SDHC.

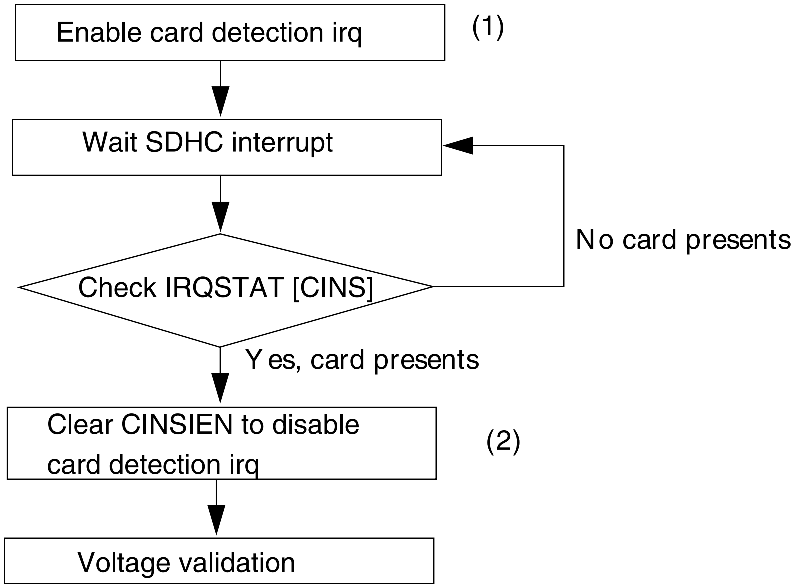


Figure 50-87. Flow diagram for card detection

Here is the card detection sequence:

- Set the CINSIEN bit to enable card detection interrupt
- When an interrupt from the SDHC is received, check IRQSTAT[CINS] in the Interrupt Status register to see whether it was caused by card insertion
- Clear the CINSIEN bit to disable the card detection interrupt and ignore all card insertion interrupts afterwards

50.6.2.2 Reset

The host consists of three types of resets:

- Hardware reset (card and host) which is driven by power on reset (POR)
- Software reset (host only) is proceed by the write operation on the SYSCTL[RSTD], SYSCTL[RSTC], or SYSCTL[RSTA] bits to reset the data part, command part, or all parts of the host controller, respectively
- Card reset (card only). The command, Go_Idle_State (CMD0), is the software reset command for all types of MMC cards, SD Memory cards. This command sets each card into the idle state regardless of the current card state. For an SD I/O Card, CMD52 is used to write an I/O reset in the CCCR. The cards are initialized with a default relative card address (RCA = 0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. The following diagram illustrates the software flow to reset both the SDHC and the card.

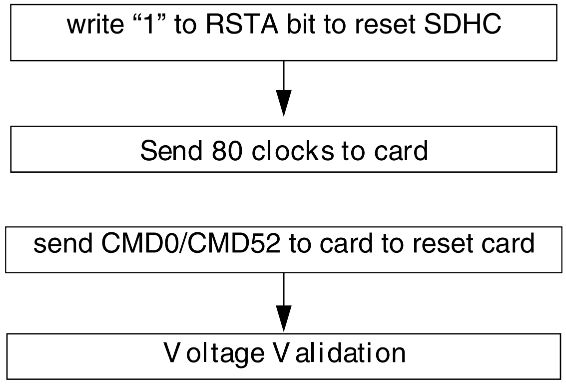


Figure 50-88. Flow chart for reset of the SDHC and SD I/O card

```
software_reset()
{
set_bit(SYSCTRL, RSTA); // software reset the Host
set DTOCV and SDCLKFS bit fields to get the SD_CLK of frequency around 400kHz
configure IO pad to set the power voltage of external card to around 3.0V
poll bits CIHB and CDIHB bits of PRSSTAT to wait both bits are cleared
set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
send_command(CMD_GO_IDLE_STATE, <other parameters>); // reset the card with CMD0
or send_command(CMD_IO_RW_DIRECT, <other parameters>);
}
```

50.6.2.3 Voltage validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification. However, the supported minimum and maximum values for V_{DD} are defined in the Operation Conditions Register (OCR) and may not cover the whole range. Cards that store the CID and CSD data in the preload memory are able to communicate this information only under data transfer V_{DD} conditions. This means if the host and card have noncommon V_{DD} ranges, the card will not be able to complete the identification cycle, nor will it be able to send CSD data.

Therefore, a special command Send_Op_Cont (CMD1 for MMC), SD_Send_Op_Cont (ACMD41 for SD Memory) and IO_Send_Op_Cont (CMD5 for SD I/O) is used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards which do not match the V_{DD} range(s) desired by the host. This is accomplished by the host sending the desired V_{DD} voltage window as the operand of this command. Cards that can't perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive state. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before

sending out-of-range cards into the inactive state. This query should be used if the host is able to select a common voltage range or if a notification shall be sent to the system when a nonusable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_arguement)
{
label the card as UNKNOWN;
send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO
operation voltage, command argument is zero
if (RESP_TIMEOUT != wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted
 if (0 < number of IO functions) {
  label the card as SDIO;
  IORDY = 0;
  while (!(IORDY in IO OCR response)) { // set voltage range for each IO function
   send_command(IO_SEND_OP_COND, <voltage range>, <other parameter>);
   wait_for_response(IO_SEND_OP_COND);
  } // end of while ...
 } // end of if (0 < ...
 if (memory part is present inside SDIO card) Label the card as SDCombo; // this is an
SD-Combo card
} // end of if (RESP_TIMEOUT ...
if (the card is labelled as SDIO card) return; // card type is identified and voltage range
is
set, so exit the function;
send_command(APP_CMD, 0x0, <other parameters are omitted>); // CMD55, Application specific
CMD
prefix
if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55 is accepted
 send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41, to set voltage range
for memory part or SD card
 wait_for_response(SD_APP_OP_COND); // voltage range is set
 if (card type is UNKNOWN) label the card as SD;
 return; //
} // end of if (no error ...
else if (errors other than time-out occur) { // command/response pair is corrupted
 deal with it by program specific manner;
} // of else if (response time-out
else { // CMD55 is refuse, it must be MMC card or CE-ATA card
 if (card is already labelled as SDCombo) { // change label
  re-label the card as SDIO;
  ignore the error or report it;
  return; // card is identified as SDIO card
 } // of if (card is ...
 send_command(SEND_OP_COND, <voltage range>, <...>);
 if (RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted, either
  label the card as UNKNOWN;
  return;
 } // of if (RESP_TIMEOUT ...
 if (check for CE-ATA signature succeeded) { // the card is CE-ATA
  store CE-ATA specific info from the signature;
  label the card as CE-ATA;
 } // of if (check for CE-ATA ...
 else label the card as MMC;
} // of else
}
```

50.6.2.4 Card registry

Card registry for the MMC and SD/SDIO/SD combo cards are different.

For the SD card, the identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V, as defined by the card specification. At this time, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host will request the card to send their valid operation conditions. The response to ACMD41 is the operation condition register of the card. The same command shall be send to all of the new cards in the system. Incompatible cards are put into the inactive state. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified, in the ready state, send their CID number as the response. After the CID is sent by the card, the card goes into the Identification state.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA will be used to address the card for future data transfer operations. After the RCA is received, the card changes its state to the Standby state. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in system.

For MMC operation, the host starts the card identification process in Open-Drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is activated, the host will request the cards to send their valid operation conditions (CMD1). The response to CMD1 is the wired OR operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive state.

The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards, the cards in ready state, simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle. Because the CID is unique for each card, only one card can successfully send its full CID to the host. This card then goes into the Identification state.

Thereafter, the host issues Set_Relative_Addr (CMD3) to assign to the card a relative card address (RCA). When the RCA is received the card state changes to the standby state, and the card does not react in further identification cycles, and its output driver

switches from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

For operation as MMC cards:

```
card_registry()
{
do { // decide RCA for each card until response time-out
 if(card is labelled as SDCombo or SDIO) { // for SDIO card like device
  send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO card to publish its
RCA
  retrieve RCA from response;
 } // end if (card is labelled as SDCombo ...
 else if (card is labelled as SD) { // for SD card
  send_command(ALL_SEND_CID, <...>);
  if (RESP_TIMEOUT == wait_for_response(ALL_SEND_CID)) break;
  send_command(SET_RELATIVE_ADDR, <...>);
  retrieve RCA from response;
 } // else if (card is labelled as SD ...
 else if (card is labelled as MMC or CE-ATA) { // treat CE-ATA as MMC
  send_command(ALL_SEND_CID, <...>);
  rca = 0x1; // arbitrarily set RCA, 1 here for example, this RCA is also the
relative address to access the CE-ATA card
  send_command(SET_RELATIVE_ADDR, 0x1 << 16, <...>); // send RCA at upper 16
bits
 } // end of else if (card is labelled as MMC ...
} while (response is not time-out);
}
```

50.6.3 Card access

This section discusses the various card access methods.

50.6.3.1 Block write

This section discusses the block write access methods.

50.6.3.1.1 Normal write

During a block write (CMD24 - 27, CMD60, CMD61), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card shall indicate the failure on the dat line. The transferred data will be discarded and not written, and all further transmitted blocks in Multiple Block Write mode will be ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and aborts the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in

the status register, and while ignoring all further data transfer, waits in the Receive-Data-state for a stop command. The write operation is also aborted if the host tries to write over a write protected area.

For MMC and SD cards, programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO cards at any time, and the card will respond with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 to select a different card to place the card into the Standby state and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate the busy indication by pulling DAT to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card incorporates the internal DMA and the write operation is a multi-block write with the Auto CMD12 enabled. For the other two methods, by means of external DMA or CPU polling status, with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to write to a card is:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:
    a. For SD/MMC cards, use SET_BLOCKLEN (CMD16)
    b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the eSDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the eSDHC number block register (NOB), nob is 5 (for instance).

5. Disable the buffer write ready interrupt, configure the DMA settings, and enable the eSDHC DMA when sending the command with data transfer. AC12EN should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see whether a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

50.6.3.1.2 Write with pause

The write operation can be paused during the transfer. Instead of stopping the SD_CLK at any time to pause all the operations, which is also inaccessible to the host driver, the driver can set PROCTL[SABGREQ] to pause the transfer between the data blocks. As there is no time-out condition in a write operation during the data blocks, a write to all types of cards can be paused in this way, and if the DAT0 line is not required to deassert to release the busy state, no suspend command is needed.

Like the flow described in Normal write, the write with pause is shown with the same kind of write operation:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
    a. For SD/MMC, use SET_BLOCKLEN (CMD16)
    b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the SDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the SDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer write ready interrupt, configure the DMA settings, and enable the SDHC DMA when sending the command with data transfer. The XFERTYP[AC12EN] bit should also be set.
6. Set PROCTL[SABGREQ].
7. Wait for the transfer complete interrupt.
8. Clear PROCTL[SABGREQ].
9. Check the status bit to see whether a write CRC error occurred.

10. Set PROCTL[CREQ] to continue the write operation.
11. Wait for the transfer complete interrupt.
12. Check the status bit to see whether a write CRC error occurred, or some another error, that occurred during the auto12 command sending and response receiving.

The number of blocks left during the data transfer is accessible by reading the contents of the BLKATTR[BLKCNT] . As the data transfer and the setting of the PROCTL[SABGREQ] bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The driver shall read the value of BLKATTR[BLKCNT] after the transfer is paused and the transfer complete interrupt is received.

It is also possible that the last block has begun when the stop at block gap request is sent to the buffer. In this case, the next block gap is actually the end of the transfer. These types of requests are ignored and the Driver should treat this as a non-pause transfer and deal with it as a common write operation.

When the write operation is paused, the data transfer inside the host system is not stopped, and the transfer is active until the data buffer is full. Because of this (if not needed), it is recommended to avoid using the suspend command for the SDIO card. This is because when such a command is sent, the SDHC thinks the system will switch to another function on the SDIO card, and flush the data buffer. The SDHC takes the resume command as a normal command with data transfer, and it is left for the driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, XFERTYP[MSBSEL] and XFERTYP[BCEN] are set as well as XFERTYP[AC12EN]. However, the SDHC will automatically send a CMD12 to mark the end of the multiblock transfer.

50.6.3.2 Block read

This section discusses the block read access methods.

50.6.3.2.1 Normal read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the Transfer state. For multi blocks read, data blocks will be continuously transferred until a stop command is issued.

The software flow to read from a card incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other two methods (by means of external DMA or CPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is:

1. Check the card status, wait until card is ready for data.
2. Set the card block length/size:
   a. For SD/MMC, use SET_BLOCKLEN (CMD16)
   b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O block size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
3. Set the SDHC block length register to be the same as the block length set for the card in Step 2.
4. Set the SDHC number block register (NOB), nob is 5 (for instance).
5. Disable the buffer read ready interrupt, configure the DMA settings and enable the SDHC DMA when sending the command with data transfer. XFERTYP[AC12EN] must also be set.
6. Wait for the transfer complete interrupt.
7. Check the status bit to see whether a read CRC error occurred, or some another error, occurred during the auto12 command sending and response receiving.

50.6.3.2.2 Read with pause

The read operation is not generally able to pause. Only the SDIO card and SDCombo card working under I/O mode supporting the read and wait feature can pause during the read operation. If the SDIO card support read wait (SRW bit in CCCR register is 1), the Driver can set the SABGREQ bit in the Protocol Control register to pause the transfer between the data blocks. Before setting the SABGREQ bit, make sure the RWCTL bit in the Protocol Control register is set, otherwise the eSDHC will not assert the Read Wait signal during the block gap and data corruption occurs. Set the RWCTL bit after the Read Wait capability of the SDIO card is recognized.

Like the flow described in Normal read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCR register on the SDIO card to confirm the card supports Read Wait.
2. Set RWCTL.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:
    a. For SD/MMC, use SET_BLOCKLEN (CMD16)
    b. For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7)
5. Set the SDHC block length register to be the same as the block length set for the card in Step 2.
6. Set the SDHC number block register (NOB), nob is 5 (for instance).
7. Disable the buffer read ready interrupt, configure the DMA setting and enable the eSDHC DMA when sending the command with data transfer. AC12EN must also be set.
8. Set SABGREQ.
9. Wait for the Transfer Complete interrupt.
10. Clear SABGREQ.
11. Check the status bit to see whether read CRC error occurred.
12. Set CREQ to continue the read operation.
13. Wait for the Transfer Complete interrupt.
14. Check the status bit to see whether a read CRC error occurred, or some another error, occurred during the auto12 command sending and response receiving.

Like the write operation, it is possible to meet the ending block of the transfer when paused. In this case, the SDHC will ignore the Stop At Block Gap Request and treat it as a command read operation.

Unlike the write operation, there is no remaining data inside the buffer when the transfer is paused. All data received before the pause will be transferred to the Host System. Whether the Suspend Command is sent or not, the internal data buffer is not flushed.

If the Suspend Command is sent and the transfer is later resumed by means of a Resume Command, the SDHC takes the command as a normal one accompanied with data transfer. It is left for the Driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set, as well as the AC12EN bit. However, the SDHC will automatically send the CMD12 to mark the end of multi-block transfer.

50.6.3.3 Suspend resume

The SDHC supports the suspend resume operations of SDIO cards, although slightly different than the suggested implementation of Suspend in the SDIO card specification.

50.6.3.3.1 Suspend

After setting the PROCTL[SABGREQ] bit, the host driver may send a suspend command to switch to another function of the SDIO card. The SDHC does not monitor the content of the response, therefore it doesn't know whether the suspend command succeeded or not. Accordingly, it doesn't deassert read wait for read pause. To solve this problem, the driver shall not mark the suspend command as a suspend, that is, setting the XFERTYP[CMDTYP] bits to 01. Instead, the driver shall send this command as if it were a normal command, and only when the command succeeds, and the BS bit is set in the response, can the driver send another command marked as suspend to inform the SDHC that the current transfer is suspended. As shown in the following sequence for suspend operation:

1. Set PROCTL[SABGREQ] to pause the current data transfer at block gap.
2. After IRQSTAT[BGE] is set, send the suspend command to suspend the active function. XFERTYP[CMDTYP] field must be 2'b00.
3. Check the BS bit of the CCCR in the response. If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the Driver strategy.
4. Send another normal I/O command to the suspended function. The XFERTYP[CMDTYP] of this command must be 2'b01, so the SDHC can detect this special setting and be informed that the paused operation has successfully suspended. If the paused transfer is a read operation, the SDHC stops driving DAT2 and goes to the Idle state.
5. Save the context registers in the system memory for later use, including the DMA system address register for internal DMA operation, and the block attribute register.

6. Begin operation for another function on the SDIO card.

50.6.3.3.2 Resume

To resume the data transfer, a resume command shall be issued:

1. To resume the suspended function, restore the context register with the saved value in step 5 of the suspend operation above.
2. Send the resume command. In the transfer type register, all fields are set to the value as if this were another ordinary data transfer, instead of a transfer resume, except the CMDTYP is set to 2'b10.
3. If the resume command has responded, the data transfer will be resumed.

50.6.3.4 ADMA usage

To use the ADMA in a data transfer, the host driver must prepare the correct descriptor chain prior to sending the read/write command:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4 KB address aligned).
3. If necessary, create a link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1–3.
4. Repeat steps 1–3 until all descriptors are created.
5. In the last descriptor, set the end flag to 1 and make sure the total length of all descriptors match the product of the block size and block number configured in the BLKATTR register.
6. Set the DSADDR register to the address of the first descriptor and set the PROCTL[DMAS] field to 01 to select the ADMA.
7. Issue a write or read command with XFERTYP[DMAEN] set to 1.

Steps 1–5 are independent of step 6, so step 6 can finish before steps 1–5. Regarding the descriptor configuration, do not to use the link descriptor, as it requires extra system memory access.

50.6.3.5 Transfer error

This section discusses the handling of transfer errors.

50.6.3.5.1 CRC error

It is possible at the end of a block transfer that a write CRC status error or read CRC error occurs. For this type of error, the latest block received shall be discarded. This is because the integrity of the data block is not guaranteed. Discard the following data blocks and retransfer the block from the corrupted one.

For a multi-block transfer, the host driver shall issue a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the XFERTYP[AC12EN] and BCEND bits are set, the SDHC does not automatically send a CMD12 because the last block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an auto CMD12 will be sent by the SDHC. In this case, the driver shall re-send or re-obtain the last block with a single block transfer.

50.6.3.5.2 Internal DMA error

During the data transfer with internal simple DMA, if the DMA engine encounters some error on the system bus, the DMA operation is aborted and DMA error interrupt is sent to the host system. When acknowledged by such an interrupt, the driver shall calculate the start address of data block in which the error occurs. The start address can be calculated by either:

1. Reading the DMA system address register. The error occurs during the previous burst. Taking the block size, the previous burst length and the start address of the next burst transfer into account, it is straight forward to obtain the start address of the corrupted block.
2. Reading the BLKCNT field of the block attribute register. By the number of blocks left, the total number to transfer, the start address of transfer, and the size of each block, the start address of corrupted block can be determined. When the BCEN bit is not set, the contents of the block attribute register does not change, so this method does not work.

When a DMA error occurs, abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and restart the transfer from the corrupted block to recover from the error.

50.6.3.5.3 ADMA error

There are three kinds of possible ADMA errors: transfer, invalid descriptor, and data-length mismatch. Whenever these errors occur, the DMA transfer stops and the corresponding error status bit is set. For acknowledging the status, the host driver should recover the error as shown below and retransfer from the place of interruption.

- Transfer error: May occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or the next block if no block is corrupted.
- Invalid descriptor error: For such errors, retrieve the transfer context, reset for the data part, and recreate the descriptor chain from the invalid descriptor, and issue a new transfer. As the data to transfer now may be less than the previous setting, the data length configured in the new descriptor chain should match the new value.
- Data-length mismatch error: It is similar to recover from this error. The host driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain, and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

50.6.3.5.4 Auto CMD12 error

After the last block of the multi-block transfer is sent or received, and XFERTYP[AC12EN] is set when the data transfer is initiated by the data command, the SDHC automatically sends a CMD12 to the card to stop the transfer. When errors with this command occur, the driver can deal with the situations in the following manner:

1. Auto CMD12 response time-out: It is not certain whether the command is accepted by the card or not. The driver should clear the IRQSTAT[AC12E] bits and resend the CMD12 until it is accepted by the card.
2. Auto CMD12 response CRC error: Because the card responds to the CMD12, the card will abort the transfer. The driver may ignore the error and clear the IRQSTAT[AC12E] bit.
3. Auto CMD12 conflict error or not sent: The command is not sent, so the driver shall send a CMD12 manually.

50.6.3.6 Card interrupt

The external cards can inform the host controller by means of some special signals. For the SDIO card, it can be the low level on the DAT[1] line during some special period. The SDHC only monitors the DAT[1] line and supports the SDIO interrupt.

When the SDIO interrupt is captured by the SDHC, and the host system is informed by the SDHC asserting the SDHC interrupt line, the interrupt service from the host driver is called.

Because the interrupt factor is controlled by the external card, the interrupt from the SDIO card must be served before IRQSTAT[CINT] is cleared by written 1. See Card interrupt handling for the card interrupt handling flow.

50.6.4 Switch function

MMC cards transferring data at bus widths other than 1-bit is a new feature added to the MMC specifications. The high-speed timing mode for all card devices was also recently defined in various card specifications. To enable these new features, a switch command shall be issued by the host driver.

For SDIO cards, the high-speed mode is enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed. For SD cards, the high speed mode is queried and enabled by a CMD6, with the mnemonic symbol as SWITCH_FUNC. For MMC cards , the high-speed mode is queried by a CMD8 and enabled by a CMD6, with the mnemonic symbol as SWITCH.

The 4-bit and 8-bit bus width of the MMC is also enabled by the SWITCH command, but with a different argument.

These new functions can also be disabled by a software reset. For SDIO cards it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following flowcharts do not show current capability check, which is recommended in the function switch process.

50.6.4.1 Query, enabl,e and disable SDIO high-speed mode

```
enable_sdio_high_speed_mode(void)
{
```

```
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report the SDIO card does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is
cleared;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

50.6.4.2 Query, enable, and disable SD high-speed mode

```
enable_sd_high_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0xFFFFF1 and read 64 bytes of data accompanying the R1 response;
wait data transfer done bit is set;
check if the bit 401 of received 512 bit is set;
if (bit 401 is '0') report the SD card does not support high speed mode and return;
send CMD6, with argument 0x80FFFFF1 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sd_high_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

50.6.4.3 Query, enable, and disable MMC high-speed mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support high speed mode and
return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this MMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report MMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
```

```
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into eSDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

50.6.4.4 Set MMC bus width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support multiple bit width and
return;
send CMD6 with argument 0x3B70x00; (8-bit, x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

50.6.5 ADMA operation

This section presents code examples for ADMA operation.

50.6.5.1 ADMA1 operation

```
Set_adma1_descriptor
{
      if (to start data transfer) {
          // Make sure the address is 4KB align.
          Set 'Set' type descriptor;
          {
              Set Act bits to 01;
              Set [31:12] bits data length (byte unit);
          }
          Set 'Tran' type descriptor;
          {
              Set Act bits to 10;
              Set [31:12] bits address (4KB align);
          }
      }
      else if (to fetch descriptor at non-continuous address) {
          Set Act bits to 11;
          Set [31:12] bits the next descriptor address (4KB align);
      }
      else { // other types of descriptor
          Set Act bits accordingly
      }
```

```
      if (this descriptor is the last one) {
          Set End bit to 1;
      }
      if (to generate interrupt for this descriptor) {
          Set Int bit to 1;
      }
      Set Valid bit to 1;
}
```

50.6.5.2 ADMA2 operation

```
Set_adma2_descriptor
{
      if (to start data transfer) {
          // Make sure the address is 32-bit boundary (lower 2-bit are always '00').
          Set higher 32-bit of descriptor for this data transfer initial address;
          Set [31:16] bits data length (byte unit);
          Set Act bits to '10';
      }
      else if (to fetch descriptor at non-continuous address) {
          Set Act bits to '11';
          // Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
          Set higher 32-bit of descriptor for the next descriptor address;
      }
      else { // other types of descriptor
          Set Act bits accordingly
      }
      if (this descriptor is the last one) {
          Set 'End' bit '1';
      }
      if (to generate interrupt for this descriptor) {
          Set 'Int' bit '1';
      }
      Set the 'Valid' bit to '1';
}
```

50.6.6 Fast boot operation

This section discusses fast boot operations.

50.6.6.1 Normal fast boot flow

1. Software needs to configure SYSCTL[INITA] to make sure 74 card clocks are finished.
2. Software needs to configure MMCBOOT[BOOTEN] to 1, MMCBOOT[BOOTMODE] to 0, and MMCBOOT[BOOTACK] to select the ack mode or not. If sending through DMA mode, software needs to configure MMCBOOT[AUTOSABGEN] to enable automatically stop at block gap feature, and MMCBOOT[DTOCVACK] to select the ack timeout value according to the sd clk frequence.

3. Software then needs to configure BLKATTR register to set block size/no.
4. Software needs to configure PROCTL[DTW].
5. Software needs to configure CMDARG to set argument if needed (no need in normal fast boot).
6. Software needs to configure XFERTYP register to start the boot process. In Normal Boot mode, XFERTYP[CMDINX], XFERTYP[CMDTYP], XFERTYP[RSPTYP], XFERTYP[CICEN], XFERTYP[CCCEN], XFERTYP[AC12EN], XFERTYP[BCEN] and XFERTYP[DMAEN] are kept default value. XFERTYP[DPSEL] bit is set to 1, XFERTYP[DTDSEL] is set to 1, XFERTYP[MSBSEL] is set to 1. Note XFERTYP[DMAEN] must be configured as 0 in polling mode. And if XFERTYP[BCEN] is configured as 1, better to configure BLKATTR[BLKSIZE] to the max value.
7. When the step 6 is configured, boot process will begin. Software needs to poll the data buffer ready status to read the data from buffer in time. If boot time-out happened (ack time out or the first data read time out), Interrupt will be triggered, and software need to configure MMCBOOT[]BOOTEN] to 0 to disable boot. Thus will make CMD high, and then after at least 56 clocks, it is ready to begin normal initialization process.
8. If no time out, software needs to decide the data read is finished and then configure MMCBOOT[]BOOTEN] to 0 to disable boot. This will make CMD line high and command completed asserted. After at least 56 clocks, it is ready to begin normal initialization process.
9. Reset the host and then can begin the normal process.

50.6.6.2 Alternative fast boot flow

1. Software needs to configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software needs to configure MMCBOOT [BOOTEN] to 1, MMCBOOT [BOOTMODE] to 1, and MMCBOOT [BOOTACK] to select the ack mode or not. If sending through DMA mode, it also needs to configure MMCBOOT [AUTOSABGEN] to enable automatically stop at block gap feature. And needs to configure MMCBOOT [DTOCVACK] to select the ack timeout value according to the sd clk frequence.
3. Software then needs to configure BLKATTR register to set block size/no.

4. Software needs to configure PROCTL[DTW].
5. Software needs to configure CMDARG register to set argument to 0xFFFFFFFA.
6. Software needs to configure XFERTYP register to start the boot process by CMD0 with 0xFFFFFFFA argument . In alternative boot, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN are kept default value. DPSEL bit is set to 1, DTDSEL is set to 1, MSBSEL is set to 1. Note DMAEN should be configured as 0 in polling mode. And if BCEN is configured as 1 in Polling mode, it is better to configure blk no in Bock Attributes Register to the max value.
7. When step 6 is configured, boot process will begin. Software needs to poll the data buffer ready status to read the data from buffer in time. If boot time out (ack data time out in 50ms or data time out in 1s), host will send out the interrupt and software need to send CMD0 with reset and then configure boot enable bit to 0 to stop this process. After command completed, configure MMCBOOT[BOOTEN] to 0 to disable boot. After at least 8 clocks from command completed, card is ready for identification step.
8. If no time out, software needs to decide when to stop the boot process, and send out the CMD0 with reset and then after command completed, configure MMCBOOT[BOOTEN] to stop the process. After 8 clocks from command completed, slave(card) is ready for identification step.
9. Reset the host and then begin the normal process.

50.6.6.3 Fast boot application case in DMA mode

In the boot application case, because the image destination and the image size are contained in the beginning of the image, switching DMA parameters on the fly during MMC fast boot is required.

In fast boot, host can use ADMA2 (advanced DMA2) with two destinations.

The detail flow:

1. Software needs to configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software needs to configure MMCBOOT[BOOTEN] to 1; and MMCBOOT[BOOTMODE] to 0 (normal fast boot), to 1(alternative boot); and MMCBOOT[BOOTACK] to select the ack mode or not. In DMA mode, configure MMCBOOT[AUTOSABGEN] to 1 for enable automatically stop at block gap feature. Also configure MMCBOOT[BOOTBLKCNT] to set the VAULE1 (value of

block count that need to trans first time), so that host will stop at block gap when card block counter is equal to this value. And it needs to configure MMCBOOT[DTOCVACK] to select the ack timeout value according to the sd clk frequence.

3. Software then needs to configure BLKATTR register to set block size/no. In DMA mode, it is better to set block number to the max value(16'hffff).
4. Software needs to configure PROCTL[DTW].
5. Software enables ADMA2 by configuring PROCTL[DMAS].
6. Software need to set at least three pairs ADMA2 descriptor in boot memory (that is, in IRAM, at least 6 words). The first pair descriptor define the start address (IRAM) and data length(512byte*VALUE1) of first part boot code. Software also need to set the second pair descriptor, the second start address (any value that is writable), data length is suggest to set 1~2 word (record as VAULE2). Note: the second couple desc also transfer useful data even at lease 1 word. Because our ADMA2 can't support 0 data_length data transfer descriptor.
7. Software needs to configure CMDARG register to set argument to 0xFFFFFFFA in alternative fast boot, and doesn't need to be set in normal fast boot.
8. Software needs to configure XFERTYP register to start the boot process . XFERTYP[CMDINX], XFERTYP[CMDTYP], XFERTYP[RSPTYP], XFERTYP[CICEN], XFERTYP[CCCEN], XFERTYP[AC12EN], XFERTYP[BCEN], and XFERTYP[DMAEN] are kept at default value. XFERTYP[DPSEL] bit is set to 1, XFERTYP[DTDSEL] is set to 1, XFERTYP[MSBSEL] is set to 1. XFERTYP[DMAEN] is configured as 1 in DMA mode. And if XFERTYP[BCEN] is configured as 1, better to configure blk no in BLKATTR register to the max value.
9. When step 8 is configured, boot process will begin, the first VAULE1 block number data has transfer. Software needs to poll IRQSTAT[TC] bit to determine first transfer is end. Also software needs to poll IRQSTAT[BGE] bit to determine if first transfer stop at block gap.
10. When IRQSTAT[TC] and IRQSTAT[BGE] bits are 1, . SW can analyzes the first code of VAULE1 block, initializes the new memory device, if required, and sets the third pair of descriptors to define the start address and length of the remaining part of boot code (VAULE3 the remain boot code block). Remember set the last descriptor with END.
11. Software needs to configure MMCBOOT register (offset 0xc4) again. Set MMCBOOT[BOOTEN] bit to 1; and MMCBOOT[BOOTMODE] bit to 0 (normal fast boot), to 1 (alternative boot); and MMCBOOT[BOOTACK] bit to select the ack

mode or not. In DMA mode, configure MMCBOOT[AUTOSABGEN] bit to 1 for enable automatically stop at block gap feature. Also configure MMCBOOT[BOOTBLKCNT] bit to set the (VAULE1+1+VAULE3), so that host will stop at block gap when card block counter is equal to this value. And need to configure MMCBOOT[DTOCVACK] bit to select the ack timeout value according to the sd clk frequence.

12. Software needs to clear IRQSTAT[TC] and IRQSTAT[BGE] bit. And software need to clear PROCTL[SABGREQ], and set PROCTL[CREQ] to 1 to resume the data transfer. Host will transfer the VALUE2 and VAULE3 data to the destination that is set by descriptor.
13. Software needs to poll IRQSTAT[BGE] bit to determine if the fast boot is over.

Note

1. When ADMA boot flow is started, for SDHC, it is like a normal ADMA read operation. So setting ADMA2 descriptor as the normal ADMA2 transfer.
2. Need a few words length memory to keep descriptor.
3. For the 1~2 words data in second descriptor setting, it is the useful data, so software need to deal the data due to the application case.

50.6.7 Commands for MMC/SD/SDIO

The following table lists the commands for the MMC/SD/SDIO cards.

See the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the Multimediacard:

- Broadcast commands (bc), no response
- Broadcast commands with response (bcr), response from all cards simultaneously
- Addressed (point-to-point) commands (ac), no data transfer on the DAT
- Addressed (point-to-point) data transfer commands (adtc)

Table 50-103. Commands for MMC/SD/SDIO cards

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all MMC and SD memory cards to idle state.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all MMC and SD memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3[1]	ac	[31:6] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:0] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 ~ 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer to "SD Physical Specification V1.1" for more details.
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer to "The MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:6] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.

Table continues on the next page...

Table 50-103. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:6] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:6] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state to protect the card stack against communication breakdowns.
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	Reserved				
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21-23	Reserved				
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.

Table continues on the next page...

Table 50-103. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command shall be issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP _START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP _END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GRO UP	Removes one previously selected erase group from the erase selection.

Table continues on the next page...

Table 50-103. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.
CMD39	ac	[31:0] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in the MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43~51	Reserved				
CMD52	ac	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	[31:0] stuff bits	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather that a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57~59	Reserved				

Table continues on the next page...

Table 50-103. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Resp	Abbreviation	Description
CMD60	adtc	[31] WR [30:24] stuff bits [23:16] address [15:8] stuff bits [7:0] byte count	R1b	RW_MULTIPLE_REGISTER	These registers are used to control the behavior of the device and to retrieve status information regarding the operation of the device. All Status and Control registers are WORD (32-bit) in size and are WORD aligned. CMD60 shall be used to read and write these registers.
CMD61	adtc	[31] WR [30:16] stuff bits [15:0] data unit count	R1b	RW_MULTIPLE_BLOCK	The host issues a RW_MULTIPLE_BLOCK (CMD61) to begin the data transfer.
CMD62~63	Reserved				
ACMD6[4]	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13[5]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD memory card status.
ACMD22[6]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.
ACMD23[7]	ac		R1	SET_WR_BLK_ERASE_COUNT	-
ACMD41[8]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[9]	ac		R1	SET_CLR_CARD_DETECT	-
ACMD51[10]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1. CMD3 differs for MMC and SD cards. For MMC cards, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed MMC cards and high speed SD cards. Command SWITCH_FUNC is for high speed SD cards.
3. Command SWITCH is for high speed MMC cards . The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 50-104.
4. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this module).
5. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this module).
6. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this module).
7. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this module).

8. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this module).
9. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this module).
10. ACMDs shall be preceded with the APP_CMD command. (Commands listed are used for SD only, other SD commands not listed are not supported on this module).

The Access Bits for the EXT_CSD Access Modes are shown in the following table.

Table 50-104. EXT_CSD Access Modes

Bits	Access Name	Operation
00	Command Set	The command set is changed according to the Cmd Set field of the argument
01	Set Bits	The bits in the pointed byte are set, according to the 1 bits in the Value field.
10	Clear Bits	The bits in the pointed byte are cleared, according to the 1 bits in the Value field.
11	Write Byte	The Value field is written into the pointed byte.

50.7 Software restrictions

Software for the SDHC must observe the following restrictions.

50.7.1 Initialization active

The driver cannot set SYSCTL[INITA] when any of the command line or data lines is active, so the driver must ensure both PRSSTAT[CDIHB] and PRSSTAT[CIHB] bits are cleared. To auto clear SYSCTL[INITA], SYSCTL[SDCLKEN] must be 1, otherwise no clocks can go out to the card and SYSCTL[INITA] will never clear.

50.7.2 Software polling procedure

For polling read or write, when the software begins a buffer read or write, it must access exactly the number of times as the values set in the watermark level register. Moreover, if the block size is not the times of the value in watermark level register, read and write respectively, the software must access exactly the remained number of words at the end of each block.

For example, for read operation, if the WML[RDWML] is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block number is 2, then the access times for the burst sequence in the whole transfer process must be 4, 4, 2, 4, 4, 2.

50.7.3 Suspend operation

To suspend the data transfer, the software must inform SDHC that the suspend command is successfully accepted. To achieve this, after the suspend command is accepted by the SDIO card, software must send another normal command marked as suspend command (XFERTYP[CMDTYP] bits set as 01) to inform SDHC that the transfer is suspended.

If software needs resume the suspended transfer, it should read the value in BLKATTR[BLKCNT] to save the remained number of blocks before sending the normal command marked as suspend, otherwise on sending such suspend command, SDHC will regard the current transfer as aborted and change BLKATTR[BLKCNT] to its original value, instead of keeping the remained number of blocks.

50.7.4 Data length setting

For either ADMA (ADMA1 or ADMA2) transfer, the data in the data buffer must be word aligned, so the data length set in the descriptor must be times of 4.

50.7.5 (A)DMA address setting

To configure ADMA1/ADMA2/DMA address register, when TC[IRQSTAT] is set, the register will always update itself with the internal address value to support dynamic address synchronization, so software must ensure that TC[IRQSTAT] is cleared prior to configuring ADMA1/ADMA2/DMA address register.

50.7.6 Data port access

Data port does not support parallel access. For example, during an external DMA access, it is not allowed to write any data to the data port by CPU; or during a CPU read operation, it is also prohibited to write any data to the data port, by either CPU or external DMA. Otherwise the data would be corrupted inside the SDHC buffer.

50.7.7 Change clock frequency

SDHC does not automatically gates off the card clock when the host driver changes the clock frequency. To remove possible glitch on the card clock, clear SYSCTL[SDCLKEN] when changing clock divisor value and set SYSCTL[SDCLKEN] to 1 after PRSSTAT[SDSTB] is 1 again.

50.7.8 Multi-block read

For predefined multi-block read operation, that is, the number of blocks to read has been defined by previous CMD23 for MMC, or predefined number of blocks in CMD53 for SDIO/SDCombo, or whatever multi-block read without abort command at card side, an abort command, either automatic or manual CMD12/CMD52, is still required by SDHC after the pre-defined number of blocks are done, to drive the internal start response timeout. Manually sending an abort command with XFERTYP[RSPTYP] both bits cleared is recommended.

Chapter 51
Integrated Interchip Sound (I2S) / Synchronous Audio Interface (SAI)

51.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The I^2S (or I2S) module provides a synchronous audio interface (SAI) that supports full-duplex serial interfaces with frame synchronization such as I^2S, AC97, TDM, and codec/DSP interfaces.

51.1.1 Features

- Transmitter with independent bit clock and frame sync supporting 1 data channel
- Receiver with independent bit clock and frame sync supporting 1 data channel
- Maximum Frame Size of 32 words
- Word size of between 8-bits and 32-bits
- Word size configured separately for first word and remaining words in frame
- Asynchronous 32 × 32-bit FIFO for each transmit and receive channel
- Supports graceful restart after FIFO error

51.1.2 Block diagram

The following block diagram also shows the module clocks.

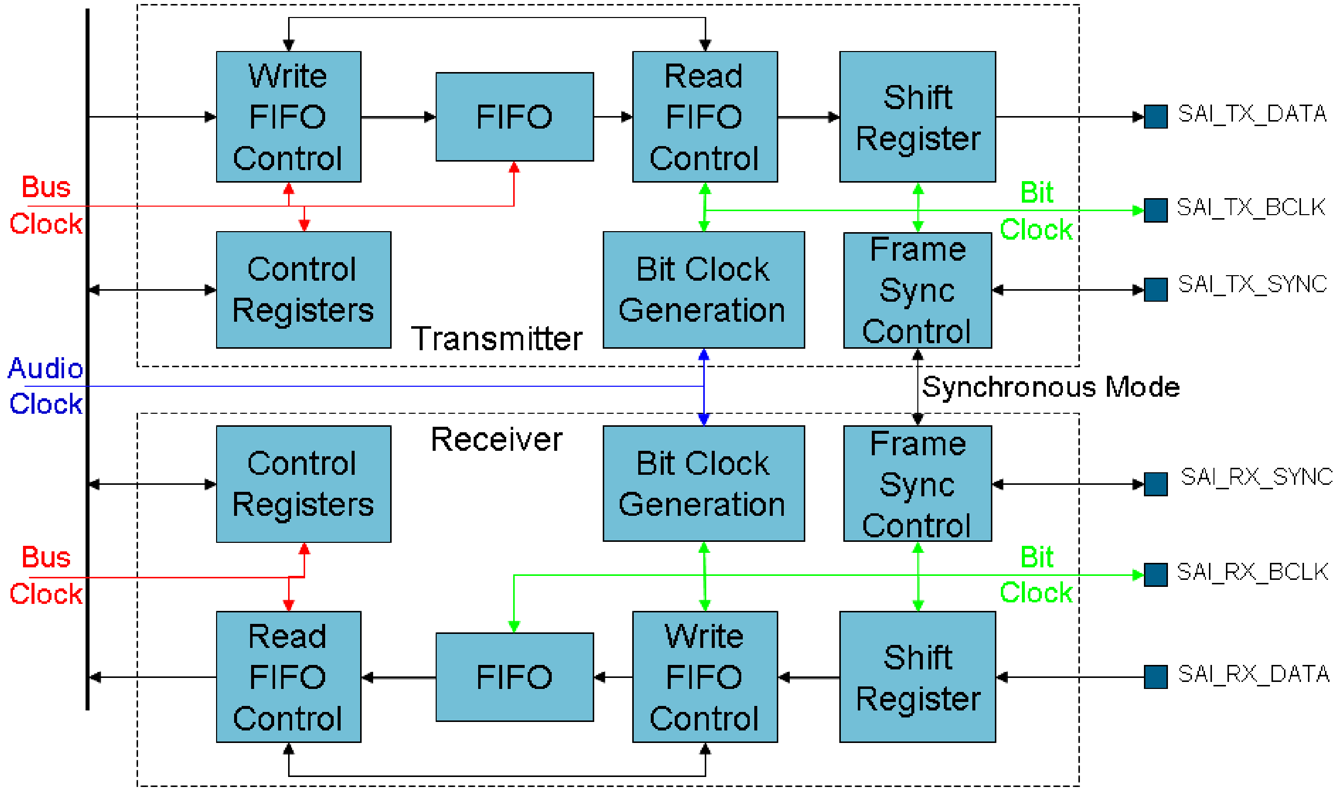


Figure 51-1. I²S/SAI block diagram

51.1.3 Modes of operation

The module operates in these MCU power modes: Run mode, stop modes, and Debug mode.

51.1.3.1 Run mode

In Run mode, the SAI transmitter and receiver operate normally.

51.1.3.2 Stop modes

In Stop mode, the transmitter is disabled after completing the current transmit frame, and, the receiver is disabled after completing the current receive frame. Entry into Stop mode is prevented–not acknowledged–while waiting for the transmitter and receiver to be disabled at the end of the current frame.

51.1.3.3 Debug mode

In Debug mode, the SAI transmitter and/or receiver can continue operating provided the Debug Enable bit is set. When TCSR[DBGE] or RCSR[DBGE] bit is clear and Debug mode is entered, the SAI is disabled after completing the current transmit or receive frame. The transmitter and receiver bit clocks are not affected by Debug mode.

51.2 External signals

Name	Function	I/O	Reset	Pull
SAI_TX_BCLK	Transmit Bit Clock	I/O	0	—
SAI_TX_SYNC	Transmit Frame Sync	I/O	0	—
SAI_TX_DATA	Transmit Data	O	0	—
SAI_RX_BCLK	Receive Bit Clock	I/O	0	—
SAI_RX_SYNC	Receive Frame Sync	I/O	0	—
SAI_RX_DATA	Receive Data	I	0	—
SAI_MCLK	Audio Master Clock	I	0	—

51.3 Memory map and register definition

A read or write access to an address after the last register will result in a bus error.

I2S memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_F000	SAI Transmit Control Register (I2S0_TCSR)	32	R/W	0000_0000h	51.3.1/2851
4002_F004	SAI Transmit Configuration 1 Register (I2S0_TCR1)	32	R/W	0000_0000h	51.3.2/2854
4002_F008	SAI Transmit Configuration 2 Register (I2S0_TCR2)	32	R/W	0000_0000h	51.3.3/2854
4002_F00C	SAI Transmit Configuration 3 Register (I2S0_TCR3)	32	R/W	0000_0000h	51.3.4/2856
4002_F010	SAI Transmit Configuration 4 Register (I2S0_TCR4)	32	R/W	0000_0000h	51.3.5/2857
4002_F014	SAI Transmit Configuration 5 Register (I2S0_TCR5)	32	R/W	0000_0000h	51.3.6/2858
4002_F020	SAI Transmit Data Register (I2S0_TDR0)	32	W (always reads 0)	0000_0000h	51.3.7/2859
4002_F040	SAI Transmit FIFO Register (I2S0_TFR0)	32	R	0000_0000h	51.3.8/2860
4002_F060	SAI Transmit Mask Register (I2S0_TMR)	32	R/W	0000_0000h	51.3.9/2860

Table continues on the next page...

I2S memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_F080	SAI Receive Control Register (I2S0_RCSR)	32	R/W	0000_0000h	51.3.10/ 2861
4002_F084	SAI Receive Configuration 1 Register (I2S0_RCR1)	32	R/W	0000_0000h	51.3.11/ 2864
4002_F088	SAI Receive Configuration 2 Register (I2S0_RCR2)	32	R/W	0000_0000h	51.3.12/ 2865
4002_F08C	SAI Receive Configuration 3 Register (I2S0_RCR3)	32	R/W	0000_0000h	51.3.13/ 2866
4002_F090	SAI Receive Configuration 4 Register (I2S0_RCR4)	32	R/W	0000_0000h	51.3.14/ 2867
4002_F094	SAI Receive Configuration 5 Register (I2S0_RCR5)	32	R/W	0000_0000h	51.3.15/ 2869
4002_F0A0	SAI Receive Data Register (I2S0_RDR0)	32	R	0000_0000h	51.3.16/ 2869
4002_F0C0	SAI Receive FIFO Register (I2S0_RFR0)	32	R	0000_0000h	51.3.17/ 2870
4002_F0E0	SAI Receive Mask Register (I2S0_RMR)	32	R/W	0000_0000h	51.3.18/ 2870
4003_0000	SAI Transmit Control Register (I2S1_TCSR)	32	R/W	0000_0000h	51.3.1/2851
4003_0004	SAI Transmit Configuration 1 Register (I2S1_TCR1)	32	R/W	0000_0000h	51.3.2/2854
4003_0008	SAI Transmit Configuration 2 Register (I2S1_TCR2)	32	R/W	0000_0000h	51.3.3/2854
4003_000C	SAI Transmit Configuration 3 Register (I2S1_TCR3)	32	R/W	0000_0000h	51.3.4/2856
4003_0010	SAI Transmit Configuration 4 Register (I2S1_TCR4)	32	R/W	0000_0000h	51.3.5/2857
4003_0014	SAI Transmit Configuration 5 Register (I2S1_TCR5)	32	R/W	0000_0000h	51.3.6/2858
4003_0020	SAI Transmit Data Register (I2S1_TDR0)	32	W (always reads 0)	0000_0000h	51.3.7/2859
4003_0040	SAI Transmit FIFO Register (I2S1_TFR0)	32	R	0000_0000h	51.3.8/2860
4003_0060	SAI Transmit Mask Register (I2S1_TMR)	32	R/W	0000_0000h	51.3.9/2860
4003_0080	SAI Receive Control Register (I2S1_RCSR)	32	R/W	0000_0000h	51.3.10/ 2861
4003_0084	SAI Receive Configuration 1 Register (I2S1_RCR1)	32	R/W	0000_0000h	51.3.11/ 2864
4003_0088	SAI Receive Configuration 2 Register (I2S1_RCR2)	32	R/W	0000_0000h	51.3.12/ 2865
4003_008C	SAI Receive Configuration 3 Register (I2S1_RCR3)	32	R/W	0000_0000h	51.3.13/ 2866
4003_0090	SAI Receive Configuration 4 Register (I2S1_RCR4)	32	R/W	0000_0000h	51.3.14/ 2867
4003_0094	SAI Receive Configuration 5 Register (I2S1_RCR5)	32	R/W	0000_0000h	51.3.15/ 2869
4003_00A0	SAI Receive Data Register (I2S1_RDR0)	32	R	0000_0000h	51.3.16/ 2869

Table continues on the next page...

I2S memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_00C0	SAI Receive FIFO Register (I2S1_RFR0)	32	R	0000_0000h	51.3.17/ 2870
4003_00E0	SAI Receive Mask Register (I2S1_RMR)	32	R/W	0000_0000h	51.3.18/ 2870
4003_1000	SAI Transmit Control Register (I2S2_TCSR)	32	R/W	0000_0000h	51.3.1/2851
4003_1004	SAI Transmit Configuration 1 Register (I2S2_TCR1)	32	R/W	0000_0000h	51.3.2/2854
4003_1008	SAI Transmit Configuration 2 Register (I2S2_TCR2)	32	R/W	0000_0000h	51.3.3/2854
4003_100C	SAI Transmit Configuration 3 Register (I2S2_TCR3)	32	R/W	0000_0000h	51.3.4/2856
4003_1010	SAI Transmit Configuration 4 Register (I2S2_TCR4)	32	R/W	0000_0000h	51.3.5/2857
4003_1014	SAI Transmit Configuration 5 Register (I2S2_TCR5)	32	R/W	0000_0000h	51.3.6/2858
4003_1020	SAI Transmit Data Register (I2S2_TDR0)	32	W (always reads 0)	0000_0000h	51.3.7/2859
4003_1040	SAI Transmit FIFO Register (I2S2_TFR0)	32	R	0000_0000h	51.3.8/2860
4003_1060	SAI Transmit Mask Register (I2S2_TMR)	32	R/W	0000_0000h	51.3.9/2860
4003_1080	SAI Receive Control Register (I2S2_RCSR)	32	R/W	0000_0000h	51.3.10/ 2861
4003_1084	SAI Receive Configuration 1 Register (I2S2_RCR1)	32	R/W	0000_0000h	51.3.11/ 2864
4003_1088	SAI Receive Configuration 2 Register (I2S2_RCR2)	32	R/W	0000_0000h	51.3.12/ 2865
4003_108C	SAI Receive Configuration 3 Register (I2S2_RCR3)	32	R/W	0000_0000h	51.3.13/ 2866
4003_1090	SAI Receive Configuration 4 Register (I2S2_RCR4)	32	R/W	0000_0000h	51.3.14/ 2867
4003_1094	SAI Receive Configuration 5 Register (I2S2_RCR5)	32	R/W	0000_0000h	51.3.15/ 2869
4003_10A0	SAI Receive Data Register (I2S2_RDR0)	32	R	0000_0000h	51.3.16/ 2869
4003_10C0	SAI Receive FIFO Register (I2S2_RFR0)	32	R	0000_0000h	51.3.17/ 2870
4003_10E0	SAI Receive Mask Register (I2S2_RMR)	32	R/W	0000_0000h	51.3.18/ 2870
4003_2000	SAI Transmit Control Register (I2S3_TCSR)	32	R/W	0000_0000h	51.3.1/2851
4003_2004	SAI Transmit Configuration 1 Register (I2S3_TCR1)	32	R/W	0000_0000h	51.3.2/2854
4003_2008	SAI Transmit Configuration 2 Register (I2S3_TCR2)	32	R/W	0000_0000h	51.3.3/2854
4003_200C	SAI Transmit Configuration 3 Register (I2S3_TCR3)	32	R/W	0000_0000h	51.3.4/2856
4003_2010	SAI Transmit Configuration 4 Register (I2S3_TCR4)	32	R/W	0000_0000h	51.3.5/2857
4003_2014	SAI Transmit Configuration 5 Register (I2S3_TCR5)	32	R/W	0000_0000h	51.3.6/2858
4003_2020	SAI Transmit Data Register (I2S3_TDR0)	32	W (always reads 0)	0000_0000h	51.3.7/2859

Table continues on the next page...

I2S memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_2040	SAI Transmit FIFO Register (I2S3_TFR0)	32	R	0000_0000h	51.3.8/2860
4003_2060	SAI Transmit Mask Register (I2S3_TMR)	32	R/W	0000_0000h	51.3.9/2860
4003_2080	SAI Receive Control Register (I2S3_RCSR)	32	R/W	0000_0000h	51.3.10/ 2861
4003_2084	SAI Receive Configuration 1 Register (I2S3_RCR1)	32	R/W	0000_0000h	51.3.11/ 2864
4003_2088	SAI Receive Configuration 2 Register (I2S3_RCR2)	32	R/W	0000_0000h	51.3.12/ 2865
4003_208C	SAI Receive Configuration 3 Register (I2S3_RCR3)	32	R/W	0000_0000h	51.3.13/ 2866
4003_2090	SAI Receive Configuration 4 Register (I2S3_RCR4)	32	R/W	0000_0000h	51.3.14/ 2867
4003_2094	SAI Receive Configuration 5 Register (I2S3_RCR5)	32	R/W	0000_0000h	51.3.15/ 2869
4003_20A0	SAI Receive Data Register (I2S3_RDR0)	32	R	0000_0000h	51.3.16/ 2869
4003_20C0	SAI Receive FIFO Register (I2S3_RFR0)	32	R	0000_0000h	51.3.17/ 2870
4003_20E0	SAI Receive Mask Register (I2S3_RMR)	32	R/W	0000_0000h	51.3.18/ 2870

51.3.1 SAI Transmit Control Register (I2Sx_TCSR)

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TE	STOPE	DBGE	BCE	0		0	SR	0			WSF	SEF	FEF	FWF	FRF
W							FR					w1c	w1c	w1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCSR field descriptions

Field	Description
31 TE	Transmitter Enable Enables/disables the transmitter. When software clears this field, the transmitter remains enabled, and this bit remains set, until the end of the current frame. 0 Transmitter is disabled. 1 Transmitter is enabled, or transmitter has been disabled and has not yet reached end of frame.
30 STOPE	Stop Enable Configures transmitter operation in Stop mode. This field is ignored and the transmitter is disabled in all stop modes. 0 Transmitter disabled in Stop mode. 1 Transmitter enabled in Stop mode.
29 DBGE	Debug Enable

Table continues on the next page...

I2Sx_TCSR field descriptions (continued)

Field	Description
	Enables/disables transmitter operation in Debug mode. The transmit bit clock is not affected by debug mode. 0 Transmitter is disabled in Debug mode, after completing the current frame. 1 Transmitter is enabled in Debug mode.
28 BCE	Bit Clock Enable Enables the transmit bit clock, separately from the TE. This field is automatically set whenever TE is set. When software clears this field, the transmit bit clock remains enabled, and this bit remains set, until the end of the current frame. 0 Transmit bit clock is disabled. 1 Transmit bit clock is enabled.
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 FR	FIFO Reset Resets the FIFO pointers. Reading this field will always return zero. FIFO pointers should only be reset when the transmitter is disabled or the FIFO error flag is set. 0 No effect. 1 FIFO reset.
24 SR	Software Reset When set, resets the internal transmitter logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. 0 No effect. 1 Software reset.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag. 0 Start of word not detected. 1 Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 Sync error not detected. 1 Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled transmit FIFO has underrun. Write a logic 1 to this field to clear this flag. 0 Transmit underrun not detected. 1 Transmit underrun detected.

Table continues on the next page...

I2Sx_TCSR field descriptions (continued)

Field	Description
17 FWF	FIFO Warning Flag Indicates that an enabled transmit FIFO is empty. 0 No enabled transmit FIFO is empty. 1 Enabled transmit FIFO is empty.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled transmit channel FIFO is less than or equal to the transmit FIFO watermark. 0 Transmit FIFO watermark has not been reached. 1 Transmit FIFO watermark has been reached.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 Disables interrupt. 1 Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 Disables interrupt. 1 Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FWDE	FIFO Warning DMA Enable

Table continues on the next page...

I2Sx_TCSR field descriptions (continued)

Field	Description
	Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.

51.3.2 SAI Transmit Configuration 1 Register (I2Sx_TCR1)

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											TFW				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR1 field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 TFW	Transmit FIFO Watermark Configures the watermark level for all enabled transmit channels.

51.3.3 SAI Transmit Configuration 2 Register (I2Sx_TCR2)

This register must not be altered when TCSR[TE] is set.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYNC		BCS	BCI	MSEL		BCP	BCD	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR2 field descriptions

Field	Description
31–30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the receiver must be configured for asynchronous operation. 00 Asynchronous mode. 01 Synchronous with receiver. 10 Synchronous with another SAI transmitter. 11 Synchronous with another SAI receiver.
29 BCS	Bit Clock Swap This field swaps the bit clock used by the transmitter. When the transmitter is configured in asynchronous mode and this bit is set, the transmitter is clocked by the receiver bit clock (SAI_RX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the transmitter continues to use the transmit frame sync (SAI_TX_SYNC). When the transmitter is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the transmitter bit clock (SAI_TX_BCLK) but use the receiver frame sync (SAI_RX_SYNC). 0 Use the normal bit clock source. 1 Swap the bit clock source.
28 BCI	Bit Clock Input When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the transmitter is delayed by the pad output delay (the transmitter is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the transmitter when this bit is set. In synchronous mode, this bit allows the transmitter to use the slave mode timing from the datasheet, while the receiver uses the master mode timing. This field has no effect when configured for an externally generated bit clock . 0 No effect. 1 Internal logic is clocked as if bit clock was externally generated.
27–26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE:** Depending on the device, some Master Clock options might not be available. See the chip configuration details for the availability and chip-specific meaning of each option. 00 Bus Clock selected. 01 Master Clock (MCLK) 1 option selected. 10 Master Clock (MCLK) 2 option selected. 11 Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 Bit clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 Bit clock is active low with drive outputs on falling edge and sample inputs on rising edge.

Table continues on the next page...

I2Sx_TCR2 field descriptions (continued)

Field	Description
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 Bit clock is generated externally in Slave mode. 1 Bit clock is generated internally in Master mode.
23–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

51.3.4 SAI Transmit Configuration 3 Register (I2Sx_TCR3)

This register must not be altered when TCSR[TE] is set.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							TCE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											WDFL				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR3 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 TCE	Transmit Channel Enable Enables the corresponding data channel for transmit operation. A channel must be enabled before its FIFO is accessed. 0 Transmit data channel N is disabled. 1 Transmit data channel N is enabled.
15–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

I2Sx_TCR3 field descriptions (continued)

Field	Description
4–0 WDFL	Word Flag Configuration Configures which word sets the start of word flag. The value written must be one less than the word number. For example, writing 0 configures the first word in the frame. When configured to a value greater than TCR4[FRSZ], then the start of word flag is never set.

51.3.5 SAI Transmit Configuration 4 Register (I2Sx_TCR4)

This register must not be altered when TCSR[TE] is set.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		0		0	0		0			0				FRSZ		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		0				SYWD				0		MF	FSE	0	FSP	FSD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR4 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 FRSZ	Frame size Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.

Table continues on the next page...

I2Sx_TCR4 field descriptions (continued)

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 MF	MSB First Configures whether the LSB or the MSB is transmitted first. 0 LSB is transmitted first. 1 MSB is transmitted first.
3 FSE	Frame Sync Early 0 Frame sync asserts with the first bit of the frame. 1 Frame sync asserts one bit before the first bit of the frame.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 Frame sync is active high. 1 Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 Frame sync is generated externally in Slave mode. 1 Frame sync is generated internally in Master mode.

51.3.6 SAI Transmit Configuration 5 Register (I2Sx_TCR5)

This register must not be altered when TCSR[TE] is set.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WNW					0			W0W					0			FBT					0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR5 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.

Table continues on the next page...

I2Sx_TCR5 field descriptions (continued)

Field	Description
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 FBT	First Bit Shifted Configures the bit index for the first bit transmitted for each word in the frame. If configured for MSB First, the index of the next bit transmitted is one less than the current bit transmitted. If configured for LSB First, the index of the next bit transmitted is one more than the current bit transmitted. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

51.3.7 SAI Transmit Data Register (I2Sx_TDR*n*)

Address: Base address + 20h offset + (4d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	TDR[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TDR*n* field descriptions

Field	Description
31–0 TDR[31:0]	Transmit Data Register The corresponding TCR3[TCE] bit must be set before accessing the channel's transmit data register. Writes to this register when the transmit FIFO is not full will push the data written into the transmit data FIFO. Writes to this register when the transmit FIFO is full are ignored.

51.3.8 SAI Transmit FIFO Register (I2Sx_TFR*n*)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

Address: Base address + 40h offset + (4d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0									WFP					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										RFP					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TFR*n* field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 WFP	Write FIFO Pointer FIFO write pointer for transmit data channel.
15–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 RFP	Read FIFO Pointer FIFO read pointer for transmit data channel.

51.3.9 SAI Transmit Mask Register (I2Sx_TMR)

This register is double-buffered and updates:

1. When TCSR[TE] is first set
2. At the end of each frame.

This allows the masked words in each frame to change from frame to frame.

Address: Base address + 60h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TWM																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TMR field descriptions

Field	Description
31–0 TWM	Transmit Word Mask Configures whether the transmit word is masked (transmit data pin tristated and transmit data not read from FIFO) for the corresponding word in the frame. 0 Word N is enabled. 1 Word N is masked. The transmit data pins are tri-stated when masked.

51.3.10 SAI Receive Control Register (I2Sx_RCSR)

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RE	STOPE	DBGE	BCE	0		0	SR	0			WSF	SEF	FEF	FWF	FRF
W							FR					w1c	w1c	w1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCSR field descriptions

Field	Description
31 RE	Receiver Enable Enables/disables the receiver. When software clears this field, the receiver remains enabled, and this bit remains set, until the end of the current frame. 0 Receiver is disabled. 1 Receiver is enabled, or receiver has been disabled and has not yet reached end of frame.
30 STOPE	Stop Enable Configures receiver operation in Stop mode. This bit is ignored and the receiver is disabled in all stop modes. 0 Receiver disabled in Stop mode. 1 Receiver enabled in Stop mode.
29 DBGE	Debug Enable Enables/disables receiver operation in Debug mode. The receive bit clock is not affected by Debug mode. 0 Receiver is disabled in Debug mode, after completing the current frame. 1 Receiver is enabled in Debug mode.
28 BCE	Bit Clock Enable Enables the receive bit clock, separately from RE. This field is automatically set whenever RE is set. When software clears this field, the receive bit clock remains enabled, and this field remains set, until the end of the current frame. 0 Receive bit clock is disabled. 1 Receive bit clock is enabled.
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 FR	FIFO Reset Resets the FIFO pointers. Reading this field will always return zero. FIFO pointers should only be reset when the receiver is disabled or the FIFO error flag is set. 0 No effect. 1 FIFO reset.
24 SR	Software Reset Resets the internal receiver logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. 0 No effect. 1 Software reset.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag.

Table continues on the next page...

I2Sx_RCSR field descriptions (continued)

Field	Description
	0 Start of word not detected. 1 Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 Sync error not detected. 1 Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled receive FIFO has overflowed. Write a logic 1 to this field to clear this flag. 0 Receive overflow not detected. 1 Receive overflow detected.
17 FWF	FIFO Warning Flag Indicates that an enabled receive FIFO is full. 0 No enabled receive FIFO is full. 1 Enabled receive FIFO is full.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled receive channel FIFO is greater than the receive FIFO watermark. 0 Receive FIFO watermark not reached. 1 Receive FIFO watermark has been reached.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 Disables interrupt. 1 Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 Disables interrupt. 1 Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts.

Table continues on the next page...

I2Sx_RCSR field descriptions (continued)

Field	Description
	0 Disables the interrupt. 1 Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.

51.3.11 SAI Receive Configuration 1 Register (I2Sx_RCR1)

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											RFW				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR1 field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 RFW	Receive FIFO Watermark Configures the watermark level for all enabled receiver channels.

51.3.12 SAI Receive Configuration 2 Register (I2Sx_RCR2)

This register must not be altered when RCSR[RE] is set.

Address: Base address + 88h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYNC		BCS	BCI	MSEL		BCP	BCD	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR2 field descriptions

Field	Description
31–30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the transmitter must be configured for asynchronous operation. 00 Asynchronous mode. 01 Synchronous with transmitter. 10 Synchronous with another SAI receiver. 11 Synchronous with another SAI transmitter.
29 BCS	Bit Clock Swap This field swaps the bit clock used by the receiver. When the receiver is configured in asynchronous mode and this bit is set, the receiver is clocked by the transmitter bit clock (SAI_TX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the receiver continues to use the receiver frame sync (SAI_RX_SYNC). When the receiver is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the receiver bit clock (SAI_RX_BCLK) but use the transmitter frame sync (SAI_TX_SYNC). 0 Use the normal bit clock source. 1 Swap the bit clock source.
28 BCI	Bit Clock Input When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the receiver is delayed by the pad output delay (the receiver is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the receiver when this bit is set. In synchronous mode, this bit allows the receiver to use the slave mode timing from the datasheet, while the transmitter uses the master mode timing. This field has no effect when configured for an externally generated bit clock . 0 No effect. 1 Internal logic is clocked as if bit clock was externally generated.

Table continues on the next page...

I2Sx_RCR2 field descriptions (continued)

Field	Description
27–26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE:** Depending on the device, some Master Clock options might not be available. See the chip configuration details for the availability and chip-specific meaning of each option. 00 Bus Clock selected. 01 Master Clock (MCLK) 1 option selected. 10 Master Clock (MCLK) 2 option selected. 11 Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 Bit Clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 Bit Clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 Bit clock is generated externally in Slave mode. 1 Bit clock is generated internally in Master mode.
23–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

51.3.13 SAI Receive Configuration 3 Register (I2Sx_RCR3)

This register must not be altered when RCSR[RE] is set.

Address: Base address + 8Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							RCE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											WDFL				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR3 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 RCE	Receive Channel Enable Enables the corresponding data channel for receive operation. A channel must be enabled before its FIFO is accessed. 0 Receive data channel N is disabled. 1 Receive data channel N is enabled.
15–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–0 WDFL	Word Flag Configuration Configures which word the start of word flag is set. The value written should be one less than the word number (for example, write zero to configure for the first word in the frame). When configured to a value greater than the Frame Size field, then the start of word flag is never set.

51.3.14 SAI Receive Configuration 4 Register (I2Sx_RCR4)

This register must not be altered when RCSR[RE] is set.

Address: Base address + 90h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			0	0		0		0			FRSZ				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SYWD					0			MF	FSE	0	FSP	FSD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR4 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

I2Sx_RCR4 field descriptions (continued)

Field	Description
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 FRSZ	Frame Size Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 MF	MSB First Configures whether the LSB or the MSB is received first. 0 LSB is received first. 1 MSB is received first.
3 FSE	Frame Sync Early 0 Frame sync asserts with the first bit of the frame. 1 Frame sync asserts one bit before the first bit of the frame.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 Frame sync is active high. 1 Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 Frame Sync is generated externally in Slave mode. 1 Frame Sync is generated internally in Master mode.

51.3.15 SAI Receive Configuration 5 Register (I2Sx_RCR5)

This register must not be altered when RCSR[RE] is set.

Address: Base address + 94h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WNW					0			W0W					0			FBT					0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR5 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 FBT	First Bit Shifted Configures the bit index for the first bit received for each word in the frame. If configured for MSB First, the index of the next bit received is one less than the current bit received. If configured for LSB First, the index of the next bit received is one more than the current bit received. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

51.3.16 SAI Receive Data Register (I2Sx_RDR*n*)

Reading this register introduces one additional peripheral clock wait state on each read.

Address: Base address + A0h offset + (4d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDR[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RDRn field descriptions

Field	Description
31–0 RDR[31:0]	Receive Data Register The corresponding RCR3[RCE] bit must be set before accessing the channel's receive data register. Reads from this register when the receive FIFO is not empty will return the data from the top of the receive FIFO. Reads from this register when the receive FIFO is empty are ignored.

51.3.17 SAI Receive FIFO Register (I2Sx_RFRn)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

Address: Base address + C0h offset + (4d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										WFP					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0									RFP					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RFRn field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–16 WFP	Write FIFO Pointer FIFO write pointer for receive data channel.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–0 RFP	Read FIFO Pointer FIFO read pointer for receive data channel.

51.3.18 SAI Receive Mask Register (I2Sx_RMR)

This register is double-buffered and updates:

1. When RCSR[RE] is first set
2. At the end of each frame

This allows the masked words in each frame to change from frame to frame.

Address: Base address + E0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RWM																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RMR field descriptions

Field	Description
31–0 RWM	Receive Word Mask Configures whether the receive word is masked (received data ignored and not written to receive FIFO) for the corresponding word in the frame. 0 Word N is enabled. 1 Word N is masked.

51.4 Functional description

51.4.1 SAI clocking

The SAI clocks include:

- The audio master clock
- The bit clock
- The bus clock

51.4.1.1 Audio master clock

The audio master clock is used to generate the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The transmitter and receiver can independently select between the bus clock and up to three audio master clocks to generate the bit clock.

The audio master clock generation and selection is chip-specific. Refer to chip-specific clocking information about how the audio master clocks are generated.

51.4.1.2 Bit clock

The SAI transmitter and receiver support asynchronous free-running bit clocks that can be generated internally from an audio master clock or supplied externally. There is also the option for synchronous bit clock and frame sync operation between the receiver and transmitter or between multiple SAI peripherals.

Externally generated bit clocks must be:

- Enabled before the SAI transmitter or receiver is enabled
- Disabled after the SAI transmitter or receiver is disabled and completes its current frames

If the SAI transmitter or receiver is using an externally generated bit clock in asynchronous mode and that bit clock is generated by an SAI that is disabled in stop mode, then the transmitter or receiver should be disabled by software before entering stop mode. This issue does not apply when the transmitter or receiver is in a synchronous mode because all synchronous SAIs are enabled and disabled simultaneously.

51.4.1.3 Bus clock

The bus clock is used by the control and configuration registers and to generate synchronous interrupts and DMA requests.

51.4.2 SAI resets

The SAI is asynchronously reset on system reset. The SAI has a software reset and a FIFO reset.

51.4.2.1 Software reset

The SAI transmitter includes a software reset that resets all transmitter internal logic, including the bit clock generation, status flags, and FIFO pointers. It does not reset the configuration registers. The software reset remains asserted until cleared by software.

The SAI receiver includes a software reset that resets all receiver internal logic, including the bit clock generation, status flags and FIFO pointers. It does not reset the configuration registers. The software reset remains asserted until cleared by software.

51.4.2.2 FIFO reset

The SAI transmitter includes a FIFO reset that synchronizes the FIFO write pointer to the same value as the FIFO read pointer. This empties the FIFO contents and is to be used after TCSR[FEF] is set, and before the FIFO is re-initialized and TCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

The SAI receiver includes a FIFO reset that synchronizes the FIFO read pointer to the same value as the FIFO write pointer. This empties the FIFO contents and is to be used after the RCSR[FEF] is set and any remaining data has been read from the FIFO, and before the RCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

51.4.3 Synchronous modes

The SAI transmitter and receiver can operate synchronously to each other.

51.4.3.1 Synchronous mode

The SAI transmitter and receiver can be configured to operate with synchronous bit clock and frame sync.

If the transmitter bit clock and frame sync are to be used by both the transmitter and receiver:

- The transmitter must be configured for asynchronous operation and the receiver for synchronous operation.
- In synchronous mode, the receiver is enabled only when both the transmitter and receiver are enabled.
- It is recommended that the transmitter is the last enabled and the first disabled.

If the receiver bit clock and frame sync are to be used by both the transmitter and receiver:

- The receiver must be configured for asynchronous operation and the transmitter for synchronous operation.
- In synchronous mode, the transmitter is enabled only when both the receiver and transmitter are both enabled.
- It is recommended that the receiver is the last enabled and the first disabled.

When operating in synchronous mode, only the bit clock, frame sync, and transmitter/receiver enable are shared. The transmitter and receiver otherwise operate independently, although configuration registers must be configured consistently across both the transmitter and receiver.

51.4.4 Frame sync configuration

When enabled, the SAI continuously transmits and/or receives frames of data. Each frame consists of a fixed number of words and each word consists of a fixed number of bits. Within each frame, any given word can be masked causing the receiver to ignore that word and the transmitter to tri-state for the duration of that word.

The frame sync signal is used to indicate the start of each frame. A valid frame sync requires a rising edge (if active high) or falling edge (if active low) to be detected and the transmitter or receiver cannot be busy with a previous frame. A valid frame sync is also ignored (slave mode) or not generated (master mode) for the first four bit clock cycles after enabling the transmitter or receiver.

The transmitter and receiver frame sync can be configured independently with any of the following options:

- Externally generated or internally generated
- Active high or active low
- Assert with the first bit in frame or asserts one bit early
- Assert for a duration between 1 bit clock and the first word length
- Frame length from 1 to 32 words per frame
- Word length to support 8 to 32 bits per word
  - First word length and remaining word lengths can be configured separately
- Words can be configured to transmit/receive MSB first or LSB first

These configuration options cannot be changed after the SAI transmitter or receiver is enabled.

51.4.5 Data FIFO

Each transmit and receive channel includes a FIFO of size 32 × 32-bit. The FIFO data is accessed using the SAI Transmit/Receive Data Registers.

51.4.5.1 Data alignment

Data in the FIFO can be aligned anywhere within the 32-bit wide register through the use of the First Bit Shifted configuration field, which selects the bit index (between 31 and 0) of the first bit shifted.

Examples of supported data alignment and the required First Bit Shifted configuration are illustrated in Figure 51-112 for LSB First configurations and Figure 51-113 for MSB First configurations.

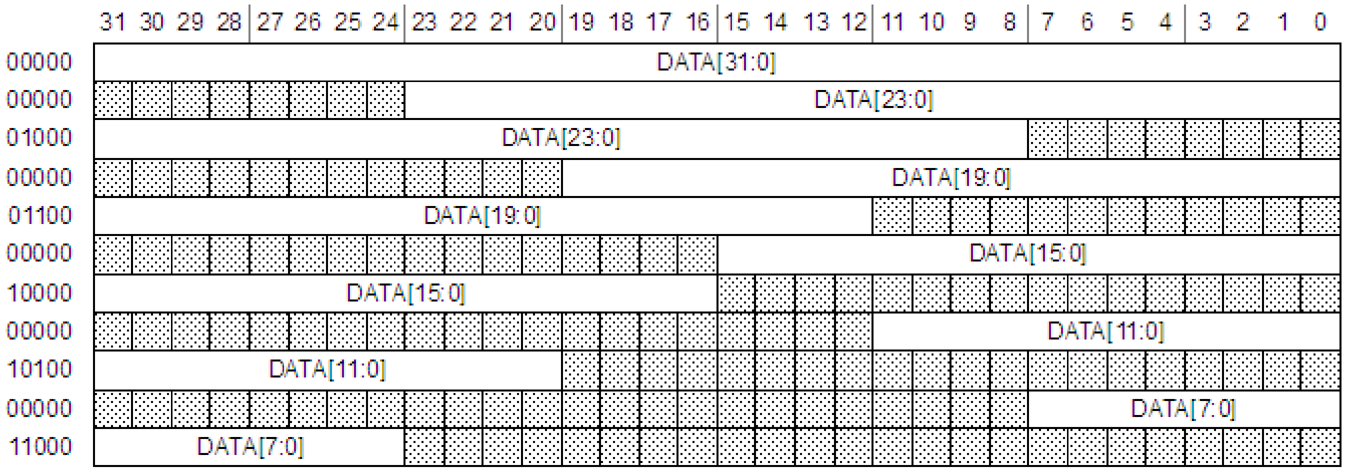


Figure 51-112. SAI first bit shifted, LSB first

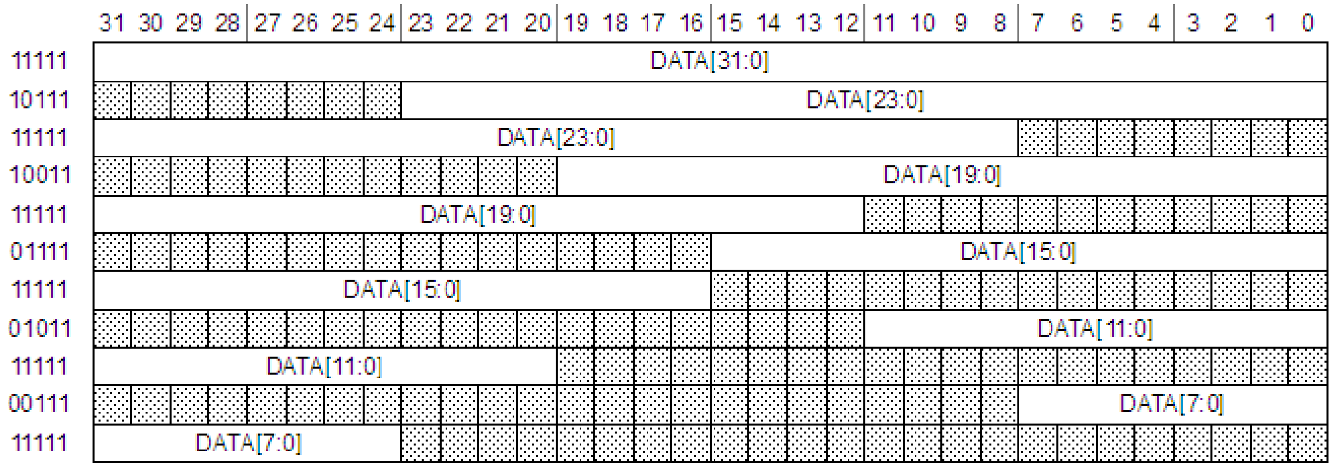


Figure 51-113. SAI first bit shifted, MSB first

51.4.5.2 FIFO pointers

When writing to a TDR, the WFP of the corresponding TFR increments after each valid write. The SAI supports 8-bit, 16-bit and 32-bit writes to the TDR and the FIFO pointer will increment after each individual write. Note that 8-bit writes should only be used when transmitting up to 8-bit data and 16-bit writes should only be used when transmitting up to 16-bit data.

Writes to a TDR are ignored if the corresponding bit of TCR3[TCE] is clear or if the FIFO is full. If the Transmit FIFO is empty, the TDR must be written at least three bit clocks before the start of the next unmasked word to avoid a FIFO underrun.

When reading an RDR, the RFP of the corresponding RFR increments after each valid read. The SAI supports 8-bit, 16-bit and 32-bit reads from the RDR and the FIFO pointer will increment after each individual read. Note that 8-bit reads should only be used when receiving up to 8-bit data and 16-bit reads should only be used when receiving up to 16-bit data.

Reads from an RDR are ignored if the corresponding bit of RCR3[RCE] is clear or if the FIFO is empty. If the Receive FIFO is full, the RDR must be read at least three bit clocks before the end of an unmasked word to avoid a FIFO overrun.

51.4.6 Word mask register

The SAI transmitter and receiver each contain a word mask register, namely TMR and RMR, that can be used to mask any word in the frame. Because the word mask register is double buffered, software can update it before the end of each frame to mask a particular word in the next frame.

The TMR causes the Transmit Data pin to be tri-stated for the length of each selected word and the transmit FIFO is not read for masked words.

The RMR causes the received data for each selected word to be discarded and not written to the receive FIFO.

51.4.7 Interrupts and DMA requests

The SAI transmitter and receiver generate separate interrupts and separate DMA requests, but support the same status flags.

51.4.7.1 FIFO request flag

The FIFO request flag is set based on the number of entries in the FIFO and the FIFO watermark configuration.

The transmit FIFO request flag is set when the number of entries in any of the enabled transmit FIFOs is less than or equal to the transmit FIFO watermark configuration and is cleared when the number of entries in each enabled transmit FIFO is greater than the transmit FIFO watermark configuration.

The receive FIFO request flag is set when the number of entries in any of the enabled receive FIFOs is greater than the receive FIFO watermark configuration and is cleared when the number of entries in each enabled receive FIFO is less than or equal to the receive FIFO watermark configuration.

The FIFO request flag can generate an interrupt or a DMA request.

51.4.7.2 FIFO warning flag

The FIFO warning flag is set based on the number of entries in the FIFO.

The transmit warning flag is set when the number of entries in any of the enabled transmit FIFOs is empty and is cleared when the number of entries in each enabled transmit FIFO is not empty.

The receive warning flag is set when the number of entries in any of the enabled receive FIFOs is full and is cleared when the number of entries in each enabled receive FIFO is not full.

The FIFO warning flag can generate an Interrupt or a DMA request.

51.4.7.3 FIFO error flag

The transmit FIFO error flag is set when the any of the enabled transmit FIFOs underflow. After it is set, all enabled transmit channels repeat the last valid word read from the transmit FIFO until TCSR[FEF] is cleared and the next transmit frame starts. All enabled transmit FIFOs must be reset and initialized with new data before TCSR[FEF] is cleared.

RCSR[FEF] is set when the any of the enabled receive FIFOs overflow. After it is set, all enabled receive channels discard received data until RCSR[FEF] is cleared and the next next receive frame starts. All enabled receive FIFOs should be emptied before RCSR[FEF] is cleared.

The FIFO error flag can generate only an interrupt.

51.4.7.4 Sync error flag

The sync error flag, TCSR[SEF] or RCSR[SEF], is set when configured for an externally generated frame sync and the external frame sync asserts when the transmitter or receiver is busy with the previous frame. The external frame sync assertion is ignored and the sync error flag is set. When the sync error flag is set, the transmitter or receiver continues checking for frame sync assertion when idle or at the end of each frame.

The sync error flag can generate an interrupt only.

51.4.7.5 Word start flag

The word start flag is set at the start of the second bit clock for the selected word, as configured by the Word Flag register field.

The word start flag can generate an interrupt only.

Chapter 52
Enhanced Serial Audio Interface (ESAI)

52.1 Overview

The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices, including industry-standard codecs, Sony/Phillips Digital Interface (SPDIF) transceivers, and other DSPs.

The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. It is a superset of the 56300 Family ESSI peripheral and of the 56000 Family SAI peripheral.

All serial transfers in the module are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is similar in that it is also intended for periodic transfers; however, it supports up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available.

The following figure shows the ESAI block diagram.

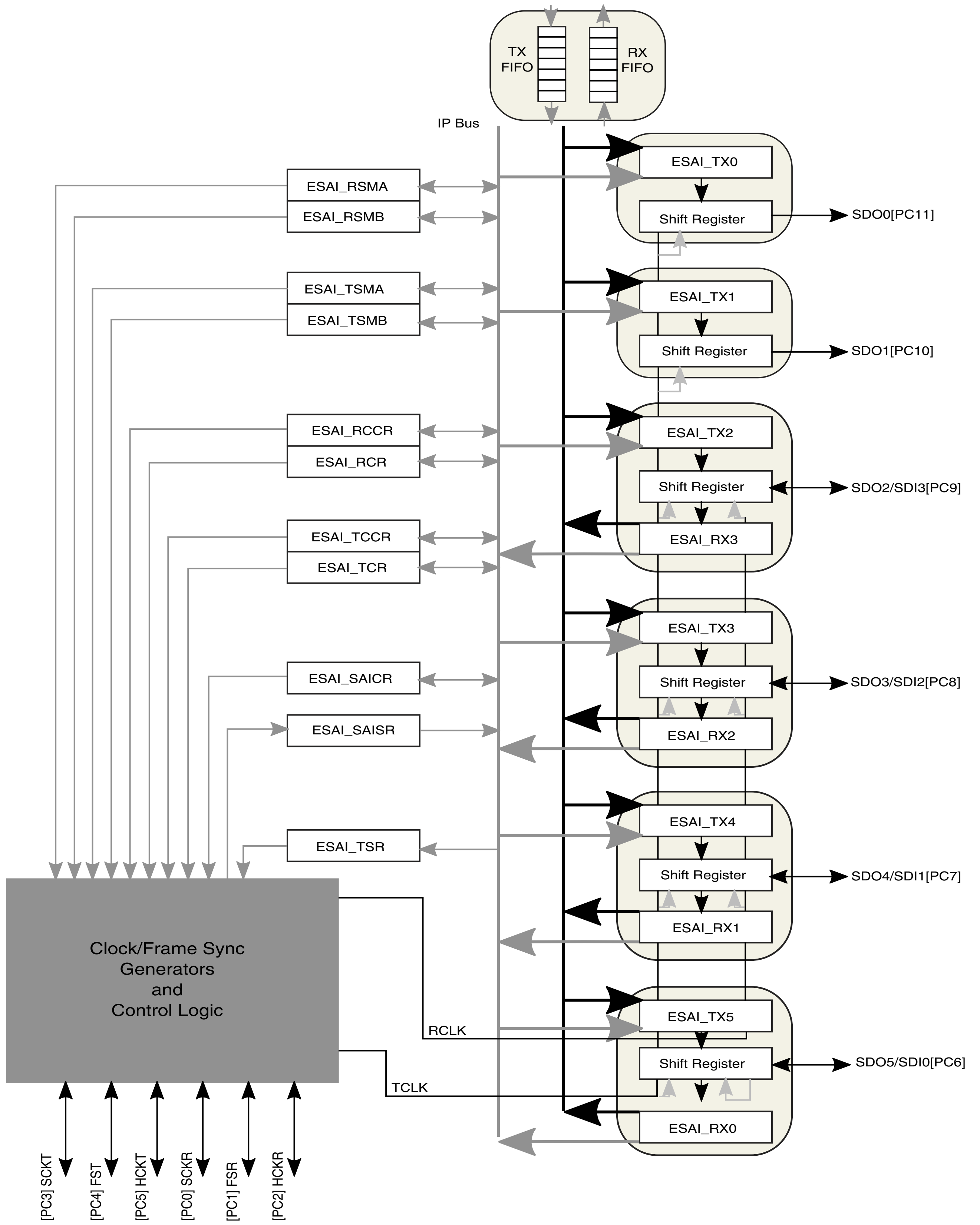


Figure 52-1. ESAI Block Diagram

52.1.1 Features

- Independent (asynchronous mode) or shared (synchronous mode) transmit and receive sections with separate or shared internal/external clocks and frame syncs, operating in Master or Slave mode.
- Up to six transmitters and four receivers with SDO2/SDI3, SDO3/SDI2, SDO4/SDI1 and SDO5/SDI0 pins shared by transmitters 2 to 5 and receivers 0 to 3. SDO0 and SDO1 pins are used by transmitters 0 and 1 only.
- Programmable data interface modes such as I2S, LSB aligned, MSB aligned
- Programmable word length (8, 12, 16, 20 or 24bits)
- Flexible selection between system clock or external oscillator as input clock source, programmable internal clock divider and frame sync generation
- AC97 support
- Time Slot Mask Registers for reduced ARM platform overhead (for both Transmit and Receive)
- 128-word Transmit FIFO shared by six transmitters
- 128-word Receive FIFO shared by four receivers

52.1.2 Modes of Operation

ESAI has three basic operating modes and many data/operation formats.

ESAI operating mode are selected by the ESAI control registers (ESAI_TCCR, ESAI_TCR, ESAI_RCCR, ESAI_RCR, and ESAI_SAICR). The main operating modes are described in the following section.

52.1.2.1 Normal/Network/On-Demand Mode Selection

Selecting between the normal mode and network mode is accomplished by clearing or setting the TMOD0-TMOD1 bits in the ESAI_TCR register for the transmitter section, as well as in the RMOD0-RMOD1 bits in the ESAI_RCR register for the receiver section.

For normal mode, the ESAI functions with one data word of I/O per frame (per enabled transmitter or receiver). The normal mode is typically used to transfer data to or from a single device.

For the network mode, 2 to 32 time slots per frame may be selected. During each frame, 0 to 32 data words of I/O may be received or transmitted. In either case, the transfers are periodic. The frame sync signal indicates the first time slot in the frame. Network mode is typically used in time division multiplexed (TDM) networks of codecs, DSPs with multiple words per frame, or multi-channel devices.

Selecting the network mode and setting the frame rate divider to zero (DC=00000) selects the on-demand mode. This special case does not generate a periodic frame sync. A frame sync pulse is generated only when data is available to transmit. The on-demand mode requires that the transmit frame sync be internal (output) and the receive frame sync be external (input). Therefore, for simplex operation, the synchronous mode could be used; however, for full-duplex operation, the asynchronous mode must be used. Data transmission that is data driven is enabled by writing data into each TX. Although the ESAI is double buffered, only one word can be written to each TX, even if the transmit shift register is empty. The receive and transmit interrupts function as usual using TDE and RDF; however, transmit underruns are impossible for on-demand transmission and are disabled.

52.1.2.2 Synchronous/Asynchronous Operating Modes

The transmit and receive sections of the ESAI may be synchronous or asynchronous, that is, the transmitter and receiver sections may use common clock and synchronization signals (synchronous operating mode), or they may have their own separate clock and sync signals (asynchronous operating mode).

The SYN bit in the ESAI_SAICR register selects synchronous or asynchronous operation. Because the ESAI is designed to operate either synchronously or asynchronously, separate receive and transmit interrupts are provided.

When SYN is cleared, the ESAI transmitter and receiver clocks and frame sync sources are independent. If SYN is set, the ESAI transmitter and receiver clocks and frame sync come from the transmitter section (either external or internal sources).

Data clock and frame sync signals can be generated internally by the ARM Core or may be obtained from external sources. If internally generated, the ESAI clock generator is used to derive high frequency clock, bit clock and frame sync signals from the ARM Core internal system clock.

52.1.2.3 Frame Sync Selection

The frame sync can be either a bit-long or word-long signal.

The transmitter frame format is defined by the TFSL bit in the ESAI_TCR register. The receiver frame format is defined by the RFSL bit in the ESAI_RCR register.

1. In the word-long frame sync format, the frame sync signal is asserted during the entire word data transfer period. This frame sync length is compatible with codecs, SPI serial peripherals, serial A/D and D/A converters, shift registers and telecommunication PCM serial I/O.
2. In the bit-long frame sync format, the frame sync signal is asserted for one bit clock immediately before the data transfer period. This frame sync length is compatible with Intel and National components, codecs and telecommunication PCM serial I/O.

The relative timing of the word length frame sync as referred to the data word is specified by the TFSR bit in the ESAI_TCR register for the transmitter section and by the RFSR bit in the ESAI_RCR register for the receive section. The word length frame sync may be generated (or expected) with the first bit of the data word, or with the last bit of the previous word. TFSR and RFSR are ignored when a bit length frame sync is selected.

Polarity of the frame sync signal may be defined as positive (asserted high) or negative (asserted low). The TFSP bit in the ESAI_TCCR register specifies the polarity of the frame sync for the transmitter section. The RFSP bit in the ESAI_RCCR register specifies the polarity of the frame sync for the receiver section.

The ESAI receiver looks for a receive frame sync leading edge (trailing edge if RFSP is set) only when the previous frame is completed. If the frame sync goes high before the frame is completed (or before the last bit of the frame is received in the case of a bit frame sync or a word length frame sync with RFSR set), the current frame sync is not recognized, and the receiver is internally disabled until the next frame sync. Frames do not have to be adjacent, that is, a new frame sync does not have to immediately follow the previous frame. Gaps of arbitrary periods can occur between frames. Enabled transmitters are tri-stated during these gaps.

When operating in the synchronous mode (SYN=1), all clocks including the frame sync are generated by the transmitter section.

52.1.2.4 Shift Direction Selection

Some data formats, such as those used by codecs, specify MSB first while other data formats, such as the AES-EBU digital audio interface, specify LSB first.

The MSB/LSB first selection is made by programming RSHFD bit in the ESAI_RCR register for the receiver section and by programming the TSHFD bit in the ESAI_TCR register for the transmitter section.

52.2 External Signals

Three to twelve pins are required for operation, depending on the operating mode selected and the number of transmitters and receivers enabled.

The SDO0 and SDO1 pins are used by transmitters 0 and 1 only. The SDO2/SDI3, SDO3/SDI2, SDO4/SDI1 and SDO5/SDI0 pins are shared by transmitters 2 to 5 with receivers 0 to 3. The actual mode of operation is selected under software control. All transmitters operate fully synchronized under control of the same transmitter clock signals. All receivers operate fully synchronized under control of the same receiver clock signals.

Table 52-1. ESAI External Signals

Signal	Description	Pad	Mode	Direction
ESAI_RX_CLK	RX serial bit clock for the ESAI interface. The direction can be programmed.	ENET_MDIO	ALT2	IO
		GPIO_1	ALT0	
ESAI_RX_FS	RX frame sync signal for the ESAI interface.	ENET_REF_CLK	ALT2	IO
		GPIO_9	ALT0	
ESAI_RX_HF_CLK	RX high frequency clock for the ESAI interface.	ENET_RX_ER	ALT2	IO
		GPIO_3	ALT0	
ESAI_TX0	Used for transmitting data from the ESAI_TX0 serial transmit shift register.	GPIO_17	ALT0	IO
		NANDF_CS2	ALT2	
ESAI_TX1	Used for transmitting data from the ESAI_TX1 serial transmit shift register.	GPIO_18	ALT0	IO
		NANDF_CS3	ALT2	
ESAI_TX2_RX3	Used as TX2 for transmitting data from the ESAI_TX2 serial transmit shift register when programmed as a transmitter pin Used as the RX3 signal for receiving serial data to the ESAI_RX3 serial receive shift register when programmed as a receiver pin	ENET_TXD1	ALT2	IO
		GPIO_5	ALT0	
ESAI_TX3_RX2	Used as TX3 for transmitting data from the ESAI_TX2 serial transmit shift register when programmed as a transmitter pin Used as the RX2 signal for receiving serial data to the ESAI_RX3 serial receive shift register when programmed as a receiver pin	ENET_TX_EN	ALT2	IO
		GPIO_16	ALT0	

Table continues on the next page...

Table 52-1. ESAI External Signals (continued)

Signal	Description	Pad	Mode	Direction
ESAI_TX4_RX1	Used as TX4 for transmitting data from the ESAI_TX2 serial transmit shift register when programmed as a transmitter pin Used as the RX1 signal for receiving serial data to the ESAI_RX3 serial receive shift register when programmed as a receiver pin	ENET_TXD0	ALT2	IO
		GPIO_7	ALT0	
ESAI_TX5_RX0	Used as TX5 for transmitting data from the ESAI_TX2 serial transmit shift register when programmed as a transmitter pin Used as the RX0 signal for receiving serial data to the ESAI_RX3 serial receive shift register when programmed as a receiver pin	ENET_MDC	ALT2	IO
		GPIO_8	ALT0	
ESAI_TX_CLK	TX serial bit clock for the ESAI interface. The direction can be programmed.	ENET_CRS_DV	ALT2	IO
		GPIO_6	ALT0	
ESAI_TX_FS	Frame sync for both the transmitters and receivers in the synchronous mode (SYN=1) and for the transmitters only in asynchronous mode	ENET_RXD1	ALT2	IO
		GPIO_2	ALT0	
ESAI_TX_HF_CLK	TX high frequency clock for the ESAI interface.	ENET_RXD0	ALT2	IO
		GPIO_4	ALT0	

52.2.1 Serial Transmit 0 Data Pin

SDO0 is used for transmitting data from the ESAI_TX0 serial transmit shift register.

SDO0 is an output when data is being transmitted from the ESAI_TX0 shift register. In the on-demand mode with an internally generated bit clock, the SDO0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO0 may be programmed as a disconnected pin (PC11) when the ESAI SDO0 function is not being used. (See Table 52-47.)

52.2.2 Serial Transmit 1 Data Pin

SDO1 is used for transmitting data from the ESAI_TX1 serial transmit shift register.

SDO1 is an output when data is being transmitted from the ESAI_TX1 shift register. In the on-demand mode with an internally generated bit clock, the SDO1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO1 may be programmed as a disconnected pin (PC10) when the ESAI SDO1 function is not being used. (See Table 52-47.)

52.2.3 Serial Transmit 2/Receive 3 Data Pin

SDO2/SDI3 is used as the SDO2 for transmitting data from the ESAI_TX2 serial transmit shift register when programmed as a transmitter pin, or as the SDI3 signal for receiving serial data to the ESAI_RX3 serial receive shift register when programmed as a receiver pin.

SDO2/SDI3 is an input when data is being received by the ESAI_RX3 shift register. SDO2/SDI3 is an output when data is being transmitted from the ESAI_TX2 shift register. In the on-demand mode with an internally generated bit clock, the SDO2/SDI3 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO2/SDI3 may be programmed as a disconnected pin (PC9) when the ESAI SDO2 and SDI3 functions are not being used. (See Table 52-47.)

52.2.4 Serial Transmit 3/Receive 2 Data Pin

SDO3/SDI2 is used as the SDO3 signal for transmitting data from the ESAI_TX3 serial transmit shift register when programmed as a transmitter pin, or as the SDI2 signal for receiving serial data to the ESAI_RX2 serial receive shift register when programmed as a receiver pin.

SDO3/SDI2 is an input when data is being received by the ESAI_RX2 shift register. SDO3/SDI2 is an output when data is being transmitted from the ESAI_TX3 shift register. In the on-demand mode with an internally generated bit clock, the SDO3/SDI2

pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO3/SDI2 may be programmed as a disconnected pin (PC8) when the ESAI SDO3 and SDI2 functions are not being used. (See Table 52-47.)

52.2.5 Serial Transmit 4/Receive 1 Data Pin

SDO4/SDI1 is used as the SDO4 signal for transmitting data from the ESAI_TX4 serial transmit shift register when programmed as transmitter pin, or as the SDI1 signal for receiving serial data to the RX1 serial receive shift register when programmed as a receiver pin.

SDO4/SDI1 is an input when data is being received by the ESAI_RX1 shift register. SDO4/SDI1 is an output when data is being transmitted from the ESAI_TX4 shift register. In the on-demand mode with an internally generated bit clock, the SDO4/SDI1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO4/SDI1 may be programmed as a disconnected pin (PC7) when the ESAI SDO4 and SDI1 functions are not being used. (See Table 52-47.)

52.2.6 Serial Transmit 5/Receive 0 Data Pin

SDO5/SDI0 is used as the SDO5 signal for transmitting data from the ESAI_TX5 serial transmit shift register when programmed as transmitter pin, or as the SDI0 signal for receiving serial data to the ESAI_RX0 serial shift register when programmed as a receiver pin.

SDO5/SDI0 is an input when data is being received by the ESAI_RX0 shift register. SDO5/SDI0 is an output when data is being transmitted from the ESAI_TX5 shift register. In the on-demand mode with an internally generated bit clock, the SDO5/SDI0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO5/SDI0 may be programmed as a disconnected pin (PC6) when the ESAI SDO5 and SDI0 functions are not being used. (See Table 52-47.)

52.2.7 Receiver Serial Clock

SCKR is a bidirectional pin providing the receivers serial bit clock for the ESAI interface.

The direction of this pin is determined by the RCKD bit in the ESAI_RCCR register.The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1).

When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the ESAI_RCCR register. When configured as the output flag OF0, this pin reflects the value of the OF0 bit in the ESAI_SAICR register, and the data in the OF0 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When this pin is configured as the input flag IF0, the data value at the pin is stored in the IF0 bit in the ESAI_SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

SCKR may be programmed as a disconnected pin (PC0) when the ESAI SCKR function is not being used. (See Table 52-47.)

NOTE

Although the external ESAI serial clocks can be independent of and asynchronous to the internal 133 MHz ESAI system clock, the external ESAI serial clock frequency cannot exceed 133MHz/4 = 33.25 MHz and each external ESAI serial clock phase must exceed the minimum of 2 x 1/133MHz = 15.04ns.

For SCKR pin mode definitions, see Table 52-38.

The table below provides a list of asynchronous-mode receiver clock sources. For more information about EXTAL/ESAI clocking control bits (ERI, ERO), refer to ESAI Control Register (ESAI_ECR).

Table 52-2. Receiver Clock Sources (Asynchronous Mode Only)

RHCKD	RFSD	RCKD	ERI	ERO	Receiver Bit Clock Source	OUTPUTS		
0	0	0	N/A	N/A	SCKR	-	-	-
0	0	1	N/A	N/A	HCKR	-	-	SCKR
0	1	0	N/A	N/A	SCKR	-	FSR	-
0	1	1	N/A	N/A	HCKR	-	FSR	SCKR
1	0	0	0	0	SCKR	HCKR	-	-
1	0	0	0	1	SCKR	HCKR	-	-

Table continues on the next page...

Table 52-2. Receiver Clock Sources (Asynchronous Mode Only) (continued)

RHCKD	RFSD	RCKD	ERI	ERO	Receiver Bit Clock Source	OUTPUTS		
1	0	0	1	0	SCKR	HCKR	-	-
1	0	0	1	1	SCKR	HCKR	-	-
1	0	1	0	0	Fsys[1]	HCKR	-	SCKR
1	0	1	0	1	Fsys	HCKR	-	SCKR
1	0	1	1	0	EXTAL[2]	HCKR	-	SCKR
1	0	1	1	1	EXTAL	HCKR	-	SCKR
1	1	0	0	0	SCKR	HCKR	FSR	-
1	1	0	0	1	SCKR	HCKR	FSR	-
1	1	0	1	0	SCKR	HCKR	FSR	-
1	1	0	1	1	SCKR	HCKR	FSR	-
1	1	1	0	0	Fsys	HCKR	FSR	SCKR
1	1	1	0	1	Fsys	HCKR	FSR	SCKR
1	1	1	1	0	EXTAL	HCKR	FSR	SCKR
1	1	1	1	1	EXTAL	HCKR	FSR	SCKR
EXTAL is the on-chip clock source other than ESAI system 133MHz clock, and it is from esai_clk_root in CCM								

52.2.8 Transmitter Serial Clock

SCKT is a bidirectional pin providing the transmitters serial bit clock for the ESAI interface.

The direction of this pin is determined by the TCKD bit in the ESAI_TCCR register. The SCKT is a clock input or output used by all the enabled transmitters in the asynchronous mode (SYN=0) or by all the enabled transmitters and receivers in the synchronous mode (SYN=1).

The following table provides a list of asynchronous-mode transmitter clock sources.

Table 52-3. Transmitter Clock Sources (Asynchronous Mode Only)

THCKD	TFSD	TCKD	ETI	ETO	Transmitter Bit Clock Source	OUTPUTS		
0	0	0	N/A	N/A	SCKT	-	-	-
0	0	1	N/A	N/A	HCKT	-	-	SCKT
0	1	0	N/A	N/A	SCKT	-	FST	-
0	1	1	N/A	N/A	HCKT	-	FST	SCKT

Table continues on the next page...

Table 52-3. Transmitter Clock Sources (Asynchronous Mode Only) (continued)

THCKD	TFSD	TCKD	ETI	ETO	Transmitter Bit Clock Source	OUTPUTS		
1	0	0	0	0	SCKT	HCKT	-	-
1	0	0	0	1	SCKT	HCKT	-	-
1	0	0	1	0	SCKT	HCKT	-	-
1	0	0	1	1	SCKT	HCKT	-	-
1	0	1	0	0	Fsys[1]	HCKT	-	SCKT
1	0	1	0	1	Fsys	HCKT	-	SCKT
1	0	1	1	0	EXTAL[2]	HCKT	-	SCKT
1	0	1	1	1	EXTAL	HCKT	-	SCKT
1	1	0	0	0	SCKR	HCKT	FST	-
1	1	0	0	1	SCKR	HCKT	FST	-
1	1	0	1	0	SCKR	HCKT	FST	-
1	1	0	1	1	SCKR	HCKT	FST	-
1	1	1	0	0	Fsys	HCKT	FST	SCKT
1	1	1	0	1	Fsys	HCKT	FST	SCKT
1	1	1	1	0	EXTAL	HCKT	FST	SCKT
1	1	1	1	1	EXTAL	HCKT	FST	SCKT
EXTAL is the on-chip clock sources other than ESAI system 133MHz clock, and it is from esai_clk_root in CCM								

SCKT may be programmed as a disconnected pin (PC3) when the ESAI SCKT function is not being used. (See Table 52-47.)

For more information about EXTAL/ESAI clocking control bits (ETI, ETO), see ESAI Control Register (ESAI_ECR).

NOTE

Although the external ESAI serial clocks can be independent of and asynchronous to the internal 133 MHz ESAI system clock, the external ESAI serial clock frequency cannot exceed 133MHz/4 = 33.25 MHz and each external ESAI serial clock phase must exceed the minimum of 2 x 1/133MHz = 15.04ns.

52.2.9 Frame Sync for Receiver

FSR is a bidirectional pin providing the receivers frame sync signal for the ESAI interface. The direction of this pin is determined by the RFSD bit in ESAI_RCR register.

In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). For FSR pin mode definitions, see Table 52-39; for receiver clock signals, see Table 52-2.

When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the ESAI_RCCR register. When configured as the output flag OF1, this pin reflects the value of the OF1 bit in the ESAI_SAICR register, and the data in the OF1 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF1, the data value at the pin is stored in the IF1 bit in the ESAI_SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

FSR may be programmed as a disconnected pin (PC1) when the ESAI FSR function is not being used. (See Table 52-47.)

52.2.10 Frame Sync for Transmitter

FST is a bidirectional pin providing the frame sync for both the transmitters and receivers in the synchronous mode (SYN=1) and for the transmitters only in asynchronous mode

(SYN=0) (see Table 52-3). The direction of this pin is determined by the TFSD bit in the ESAI_TCR register. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external frame sync signal for the transmitters (and the receivers in synchronous mode).

FST may be programmed as a disconnected pin (PC4) when the ESAI FST function is not being used. (See Table 52-47.)

52.2.11 High Frequency Clock for Transmitter

HCKT is a bidirectional pin providing the transmitters high frequency clock for the ESAI interface.

The direction of this pin is determined by the THCKD bit in the ESAI_TCCR register. In the asynchronous mode (SYN=0), the HCKT pin operates as the high frequency clock input or output used by all enabled transmitters. In the synchronous mode (SYN=1), it operates as the high frequency clock input or output used by all enabled transmitters and receivers. When programmed as input this pin is used as an alternative high frequency

clock source to the ESAI transmitter rather than the ARM Core main clock. When programmed as output it can serve as a high frequency sample clock (to external DACs for example) or as an additional system clock (see Table 52-3).

HCKT may be programmed as a disconnected pin (PC5) when the ESAI HCKT function is not being used. (See Table 52-47.)

52.2.12 High Frequency Clock for Receiver

HCKR is a bidirectional pin providing the receivers high frequency clock for the ESAI interface.

The direction of this pin is determined by the RHCKD bit in the ESAI_RCCR register. In the asynchronous mode (SYN=0), the HCKR pin operates as the high frequency clock input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as the serial flag 2 pin. For HCKR pin mode definitions, see Table 52-40; for receiver clock signals, see Table 52-2.

When this pin is configured as serial flag pin, its direction is determined by the RHCKD bit in the ESAI_RCCR register. When configured as the output flag OF2, this pin reflects the value of the OF2 bit in the ESAI_SAICR register, and the data in the OF2 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF2, the data value at the pin is stored in the IF2 bit in the ESAI_SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

HCKR may be programmed as a disconnected pin (PC2) when the ESAI HCKR function is not being used. (See Table 52-47.)

52.2.13 Serial I/O Flags

Three ESAI pins (FSR, SCKR and HCKR) are available as serial I/O flags when the ESAI is operating in the synchronous mode (SYN=1).

Their operation is controlled by RCKD, RFSD, TEBE bits in the ESAI_RCR, ESAI_RCCR and ESAI_SAICR registers.The output data bits (OF2, OF1 and OF0) and the input data bits (IF2, IF1 and IF0) are double buffered to/from the HCKR, FSR and SCKR pins. Double buffering the flags keeps them in sync with the TX and RX data lines.

Each flag can be separately programmed. Flag 0 (SCKR pin) direction is selected by RCKD, RCKD=1 for output and RCKD=0 for input. Flag 1 (FSR pin) is enabled when the pin is not configured as external transmitter buffer enable (TEBE=0) and its direction is selected by RFSD, RFSD=1 for output and RFSD=0 for input. Flag 2 (HCKR pin) direction is selected by RHCKD, RHCKD=1 for output and RHCKD=0 for input.

When programmed as input flags, the SCKR, FSR and HCKR logic values, respectively, are latched at the same time as the first bit of the receive data word is sampled. Because the input was latched, the signal on the input flag pin (SCKR, FSR or HCKR) can change without affecting the input flag until the first bit of the next receive data word. When the received data words are transferred to the receive data registers, the input flag latched values are then transferred to the IF0, IF1 and IF2 bits in the SAISR register, where they may be read by software.

When programmed as output flags, the SCKR, FSR and HCKR logic values are driven by the contents of the OF0, OF1 and OF2 bits in the ESAI_SAICR register respectively, and they are driven when the transmit data registers are transferred to the transmit shift registers. The value on SCKR, FSR and HCKR is stable from the time the first bit of the transmit data word is transmitted until the first bit of the next transmit data word is transmitted. Software may change the OF0-OF2 values thus controlling the SCKR, FSR and HCKR pin values for each transmitted word. The normal sequence for setting output flags when transmitting data is as follows: wait for TDE (transmitter empty) to be set; first write the flags, and then write the transmit data to the transmit registers. OF0, OF1, and OF2 are double buffered so that the flag states appear on the pins when the transmit data is transferred to the transmit shift register, that is, the flags are synchronous with the data.

52.3 Functional Description

This section provides a complete functional description of the block.

52.3.1 ESAI After Reset

Hardware or software reset clears the port control register bits and the port direction control register bits, which configure all ESAI I/O pins as disconnected and both ESAI FIFOs are also in reset state.

The ESAI is in personal reset state while all ESAI pins are programmed as disconnected, and it is active only if at least one of the ESAI I/O pins is programmed as an ESAI pin.

52.3.2 ESAI Interrupt Requests

The ESAI can generate eight different interrupt requests

(ordered from the highest to the lowest priority):

1. **ESAI Receive Data with Exception Status**

   Occurs when the receive exception interrupt is enabled (REIE=1 in the RCR register), at least one of the enabled receive data registers is full (RDF=1) and a receiver overrun error has occurred (ROE=1 in the SAISR register). ROE is cleared by first reading the SAISR and then reading all the enabled receive data registers.

2. **ESAI Receive Even Data**

   Occurs when the receive even slot data interrupt is enabled (REDIE=1), at least one of the enabled receive data registers is full (RDF=1), the data is from an even slot (REDF=1) and no exception has occurred (ROE=0 or REIE=0).

   Reading all enabled receiver data registers clears RDF and REDF.

3. **ESAI Receive Data**

   Occurs when the receive interrupt is enabled (RIE=1), at least one of the enabled receive data registers is full (RDF=1), no exception has occurred (ROE=0 or REIE=0) and no even slot interrupt has occurred (REDF=0 or REDIE=0). Reading all enabled receiver data registers clears RDF.

4. **ESAI Receive Last Slot Interrupt**

   Occurs, if enabled (RLIE=1), after the last slot of the frame ended (in network mode only) regardless of the receive mask register setting. The receive last slot interrupt may be used for resetting the receive mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the receive last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum receive last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).

5. **ESAI Transmit Data with Exception Status**

   Occurs when the transmit exception interrupt is enabled (TEIE=1), at least one transmit data register of the enabled transmitters is empty (TDE=1) and a transmitter underrun error has occurred (TUE=1). TUE is cleared by first reading the SAISR and then writing to all the enabled transmit data registers, or to the TSR register.

6. **ESAI Transmit Last Slot Interrupt**

   Occurs, if enabled (TLIE=1), at the start of the last slot of the frame in network mode regardless of the transmit mask register setting. The transmit last slot interrupt may be used for resetting the transmit mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the transmit last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum transmit last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).

7. **ESAI Transmit Even Data**

   Occurs when the transmit even slot data interrupt is enabled (TEDIE=1), at least one of the enabled transmit data registers is empty (TDE=1), the slot is an even slot (TEDE=1) and no exception has occurred (TUE=0 or TEIE=0). Writing to all the TX registers of the enabled transmitters or to TSR clears this interrupt request.

8. **ESAI Transmit Data**

   Occurs when the transmit interrupt is enabled (TIE=1), at least one of the enabled transmit data registers is empty (TDE=1), no exception has occurred (TUE=0 or TEIE=0) and no even slot interrupt has occurred (TEDE=0 or TEDIE=0). Writing to all the TX registers of the enabled transmitters, or to the TSR clears this interrupt request.

52.3.3 ESAI DMA Requests from the FIFOs

The ESAI can generate two different DMA requests:

1. ESAI Transmit FIFO Empty - Asserts when the number of empty slots in the ESAI transmit FIFO exceeds the threshold programmed in the ESAI Transmit FIFO Configuration Register (TFCR). Automatically negates when the number of empty slots is less than the threshold programmed in the ESAI Transmit FIFO Configuration Register.
2. ESAI Receive FIFO Full - Asserts when the number of data words in the ESAI receive FIFO exceeds the threshold programmed in the ESAI Receive FIFO Configuration Register (RFCR). Automatically negates when the number of words is less than the threshold programmed in the ESAI Receive FIFO Configuration Register.

52.3.4 ESAI Transmit and Receive Shift Registers

52.3.4.1 ESAI Transmit Shift Registers

The transmit shift registers contain the data being transmitted

(Figure 52-2 and Figure 52-3).

Data is shifted out to the serial transmit data pins by the selected (internal/external) bit clock when the associated frame sync I/O is asserted.

The number of bits shifted out before the shift registers are considered empty and may be written to again can be 8, 12, 16, 20, 24 or 32 bits (determined by the slot length control bits in the TCR register). Data is shifted out of these registers MSB first if TSHFD=0 and LSB first if TSHFD=1.

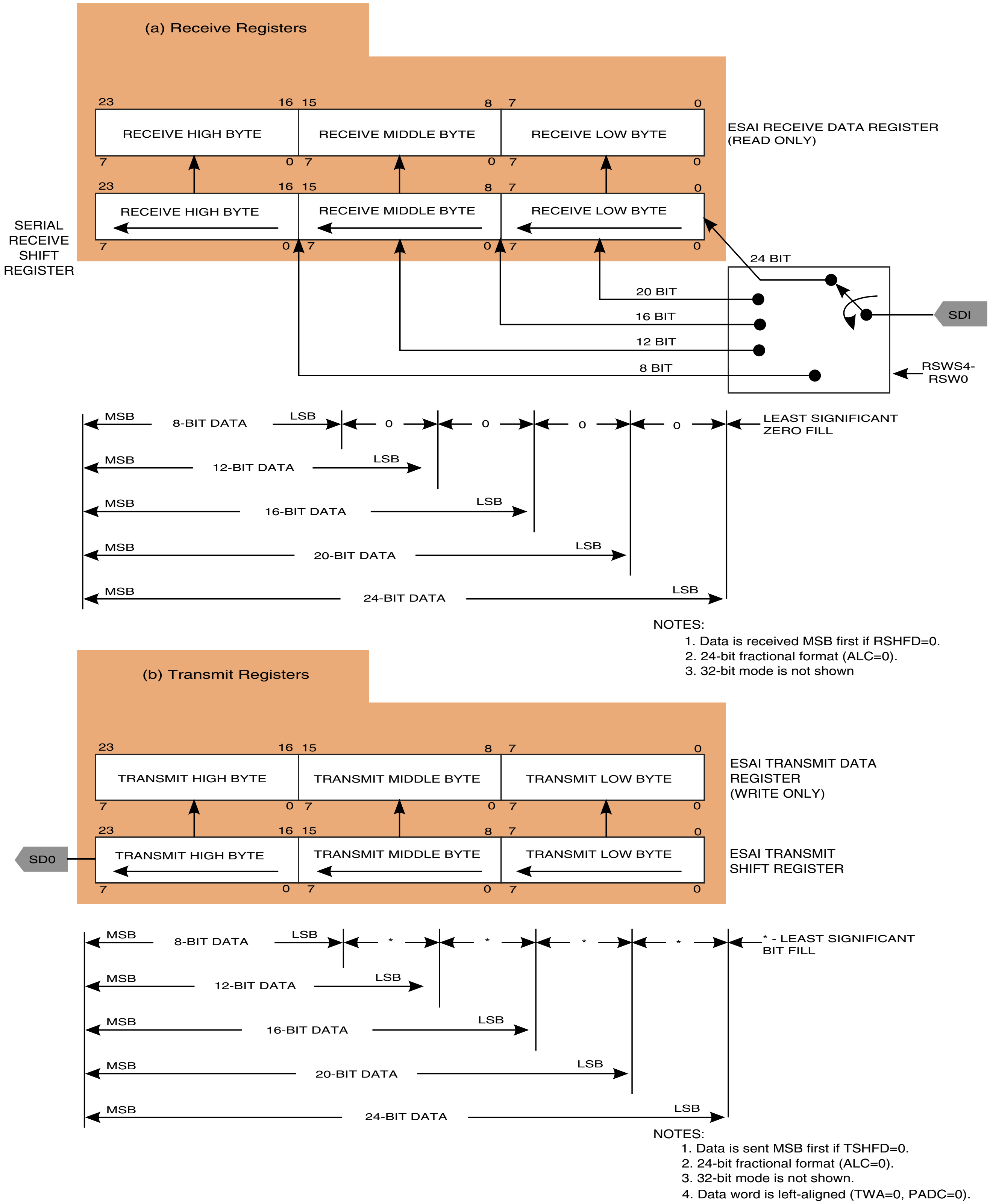


Figure 52-2. ESAI Data Path Programming Model ([R/T]SHFD=0)

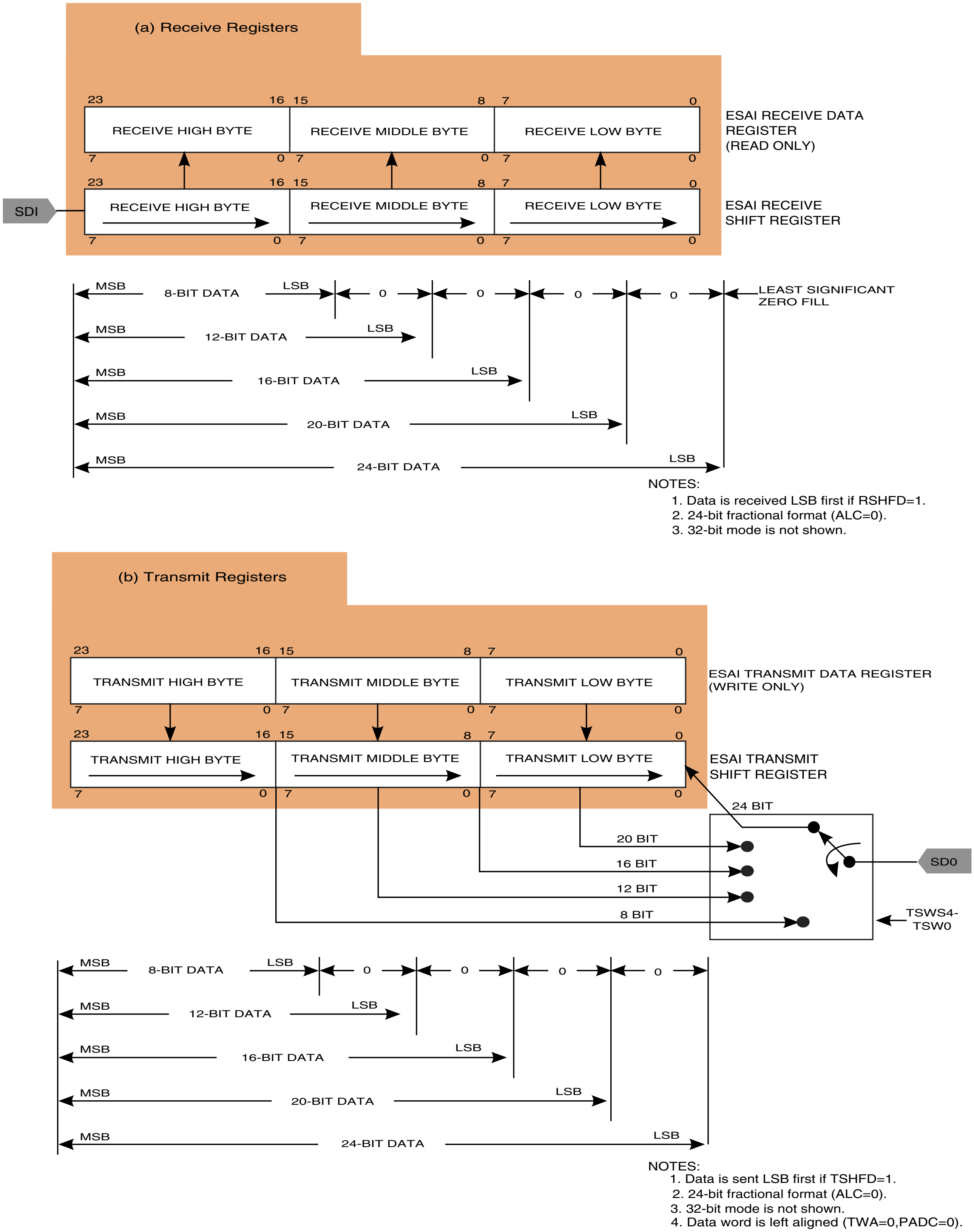


Figure 52-3. ESAI Data Path Programming Model ([R/T]SHFD=1)

52.3.4.2 ESAI Receive Shift Registers

The receive shift registers (Figure 52-2 and Figure 52-3) receive the incoming data from the serial receive data pins. Data is shifted in by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. Data is assumed to be received MSB first if RSHFD=0 and LSB first if RSHFD=1. Data is transferred to the ESAI receive data registers after 8, 12, 16, 20, 24, or 32 serial clock cycles were counted, depending on the slot length control bits in the ESAI_RCR register.

52.4 Initialization Information

52.4.1 ESAI Initialization

The correct way to initialize the ESAI is as follows:

1. Enable the ESAI logic clock by asserting bit 0 of ESAI Control Register (ESAI_ECR[0]).
2. Hardware, software, ESAI individual reset. Note that asserting bit 1 of ESAI Control Register only reset the ESAI core logic, including configuration registers, but not the ESAI FIFOs.
3. Reset ESAI FIFOs by asserting bit 1 of ESAI_TFCR and ESAI_RFCR.
4. Program ESAI control and time slot registers.(The transmit/receive enable bits of TCR/RCR should not be set.)
5. Program ESAI FIFOs via TFCR and RFCR. (Enable Transmit/Receive FIFO, enable transmitters/receivers, transmit initialization and set Transmit FIFO/Receive FIFO watermark.)
6. Write initial words to ESAI Transmit Data Register (ESAI_ETDR), at least one word per enabled transmitter slot but as many as desired. For example 4 channels with 2 slot-per-channel are enabled, then 8 words need to be written into ESAI_ETDR
7. Remove ESAI personal reset by configuring ESAI_PCRC and ESAI_PRRC.
8. Enabled Transmitters/Receivers in ESAI_TCR/ESAI_RCR.

During program execution, all ESAI pins may be defined disconnected, causing the ESAI to stop serial activity and enter the individual reset state.

All status bits of the interface are set to their reset state however, the control bits are not affected. This procedure allows the programmer to reset the ESAI separately from the other internal peripherals. During individual reset, internal DMA accesses to the data registers of the ESAI are not valid and data read is undefined.

The programmer must use an individual ESAI reset when changing the ESAI control registers (except for TEIE, REIE, TLIE, RLIE, TIE, RIE, TE0-TE5, RE0-RE3) to ensure proper operation of the interface.

NOTE

If the ESAI receiver section is already operating with some of the receivers and enabling additional receivers on the fly, that is, without first putting the ESAI receiver in the personal reset state by setting their REx control bits, it will result in erroneous data being received as the first data word for the newly enabled receivers.

52.4.2 ESAI Initialization Examples

52.4.2.1 Initializing the ESAI using Personal Reset

1. Enable the ESAI logic clock by setting bit 0 of ESAI Control Register(ESAI_ECR[0]).
2. The ESAI should be in its personal reset state (ESAI_PCRC = 0x000 and ESAI_PRRC = 0x000). In the personal reset state, both the transmitter and receiver sections of the ESAI are simultaneously reset. The TPR bit in the ESAI_TCR register may be used to reset just the transmitter section. The RPR bit in the ESAI_RCR register may be used to reset just the receiver section.
3. Configure the control registers (ESAI_TCCR, ESAI_TCR, ESAI_RCCR, ESAI_RCR) and ESAI FIFOs configuration Registers (ESAI_TFCR, ESAI_RFCR) according to the operating mode, but do not enable transmitters (TE5-TE0 = 0x0) or receivers (RE3-RE0 = 0x0). It is possible to set the interrupt enable bits which are in use during the operation (no interrupt occurs).
4. Enable the ESAI by setting the ESAI_PCRC and ESAI_PRRC register bits according to pins which are in use during operation.
5. Write initial words to ESAI Transmit Data Register (ESAI_ETDR), at least one word per enabled transmitter slot but as many as desired. For example 4 channels with 2 slot-per-channel are enabled, then 8 words need to be written into ESAI_ETDR. This step is needed even if DMA is used to service the transmitters.
6. Enable the transmitters and receivers.
7. From now on ESAI can be serviced either by polling, interrupts, or DMA.

Operation proceeds as follows:

- For internally generated clock and frame sync, these signals are active immediately after ESAI is enabled (step 4 above).

- Data is received only when one of the receive enable (REx) bits is set and after the occurrence of frame sync signal (either internally or externally generated).
- Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.

52.4.2.2 Initializing the ESAI Transmitter Section

1. It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
2. Enable the ESAI logic clock by setting bit 0 of ESAI Control Register(ESAI_ECR[0])
3. The transmitter section should be in its individual reset state (TPR = 1) and also reset the ESAI Transmit FIFO (ESAI_TFCR[1] = 1).
4. Configure the control registers ESAI_TCCR and ESAI_TCR according to the operating mode, configure the Transmit FIFO Configuration Register (bring transmit FIFO out of reset, enable Transmit FIFO, enable transmitters, transmit initialization and set watermark). Make sure to clear the transmitter enable bits (TE0-TE5). TPR must remain set.
5. Take the transmitter section out of the individual reset state by clearing TPR.
6. Write initial words to ESAI Transmit Data Register (ESAI_ETDR), at least one word per enabled transmitter slot but as many as desired. For example 4 channels with 2 slot-per-channel are enabled, then 8 words need to be written into ESAI_ETDR
7. Enable the transmitters by setting their TE bits.
8. Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.
9. From now on the transmitters are operating and can be serviced either by polling, interrupts, or DMA.

52.4.2.3 Initializing the ESAI Receiver Section

1. It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
2. Enable the ESAI logic clock by setting bit 0 of ESAI Control Register (ESAI_ECR[0])
3. The receiver section should be in its individual reset state (RPR = 1) and also reset the ESAI Receive FIFO (ESAI_RFCR[1] = 1).
4. Configure the control registers ESAI_RCCR and ESAI_RCR according to the operating mode, configure the Receive FIFO Configuration Register (bring receive

FIFO out of reset, enable Receive FIFO, receivers, and set watermark). Making sure to clear the receiver enable bits (RE0-RE3). RPR must remain set.
5. Take the receiver section out of the individual reset state by clearing RPR.
6. Enable the receivers by setting their RE bits.
7. From now on the receivers are operating and can be serviced either by polling, interrupts, or DMA.

52.5 ESAI Memory Map/Register Definition

NOTE

To access ESAI interface registers, CCM_CACRR [IP_CLK_DIV] field in CCM should be greater than or equal to 1 (divider by 2).

ESAI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_2000	ESAI Transmit Data Register (ESAI_ETDR)	32	W (always reads 0)	0000_0000h	52.5.1/2904
4006_2004	ESAI Receive Data Register (ESAI_ERDR)	32	R	0000_0000h	52.5.2/2904
4006_2008	ESAI Control Register (ESAI_ECR)	32	R/W	0000_0000h	52.5.3/2905
4006_200C	ESAI Status Register (ESAI_ESR)	32	R	0000_0000h	52.5.4/2906
4006_2010	Transmit FIFO Configuration Register (ESAI_TFCR)	32	R/W	0000_0000h	52.5.5/2907
4006_2014	Transmit FIFO Status Register (ESAI_TFSR)	32	R	0000_0000h	52.5.6/2909
4006_2018	Receive FIFO Configuration Register (ESAI_RFCR)	32	R/W	0000_0000h	52.5.7/2910
4006_201C	Receive FIFO Status Register (ESAI_RFSR)	32	R	0000_0000h	52.5.8/2911
4006_2080	Transmit Data Register n (ESAI_TX0)	32	W (always reads 0)	0000_0000h	52.5.9/2912
4006_2084	Transmit Data Register n (ESAI_TX1)	32	W (always reads 0)	0000_0000h	52.5.9/2912
4006_2088	Transmit Data Register n (ESAI_TX2)	32	W (always reads 0)	0000_0000h	52.5.9/2912
4006_208C	Transmit Data Register n (ESAI_TX3)	32	W (always reads 0)	0000_0000h	52.5.9/2912

Table continues on the next page...

ESAI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_2090	Transmit Data Register n (ESAI_TX4)	32	W (always reads 0)	0000_0000h	52.5.9/2912
4006_2094	Transmit Data Register n (ESAI_TX5)	32	W (always reads 0)	0000_0000h	52.5.9/2912
4006_2098	ESAI Transmit Slot Register (ESAI_TSR)	32	W (always reads 0)	0000_0000h	52.5.10/ 2913
4006_20A0	Receive Data Register n (ESAI_RX0)	32	R	0000_0000h	52.5.11/ 2913
4006_20A4	Receive Data Register n (ESAI_RX1)	32	R	0000_0000h	52.5.11/ 2913
4006_20A8	Receive Data Register n (ESAI_RX2)	32	R	0000_0000h	52.5.11/ 2913
4006_20AC	Receive Data Register n (ESAI_RX3)	32	R	0000_0000h	52.5.11/ 2913
4006_20CC	Serial Audio Interface Status Register (ESAI_SAISR)	32	R	0000_0000h	52.5.12/ 2914
4006_20D0	Serial Audio Interface Control Register (ESAI_SAICR)	32	R/W	0000_0000h	52.5.13/ 2917
4006_20D4	Transmit Control Register (ESAI_TCR)	32	R/W	0000_0000h	52.5.14/ 2920
4006_20D8	Transmit Clock Control Register (ESAI_TCCR)	32	R/W	0000_0000h	52.5.15/ 2927
4006_20DC	Receive Control Register (ESAI_RCR)	32	R/W	0000_0000h	52.5.16/ 2931
4006_20E0	Receive Clock Control Register (ESAI_RCCR)	32	R/W	0000_0000h	52.5.17/ 2935
4006_20E4	Transmit Slot Mask Register A (ESAI_TSMA)	32	R/W	0000_FFFFh	52.5.18/ 2938
4006_20E8	Transmit Slot Mask Register B (ESAI_TSMB)	32	R/W	0000_FFFFh	52.5.19/ 2939
4006_20EC	Receive Slot Mask Register A (ESAI_RSMA)	32	R/W	0000_FFFFh	52.5.20/ 2940
4006_20F0	Receive Slot Mask Register B (ESAI_RSMB)	32	R/W	0000_FFFFh	52.5.21/ 2941
4006_20F8	Port C Direction Register (ESAI_PRRC)	32	R/W	0000_0000h	52.5.22/ 2942
4006_20FC	Port C Control Register (ESAI_PCRC)	32	R/W	0000_0000h	52.5.23/ 2942

52.5.1 ESAI Transmit Data Register (ESAI_ETDR)

Address: 4006_2000h base + 0h offset = 4006_2000h

Bit	31–0
R	0
W	ETDR
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

ESAI_ETDR field descriptions

Field	Description
31–0 ETDR	ESAI Transmit Data Register. Writing to this register stores the data written into the ESAI Transmit FIFO. Writing to this register when the Transmit FIFO is full causes the data written to be lost (the existing data within the FIFO is not overwritten). When multiple ESAI transmitters are enabled, the data for each transmitter must be interleaved from lowest transmitter to highest transmitter (for example, if transmitters 0, 2 and 3 are enabled then data must be written as follows: transmitter #0, transmitter #2, transmitter #3, transmitter #0, transmitter #2, transmitter #3, transmitter #0, etc). Data within the ESAI Transmit FIFO is passed to the ESAI transmit shifter registers as defined by the Transmit Word Alignment configuration bits.

52.5.2 ESAI Receive Data Register (ESAI_ERDR)

Address: 4006_2000h base + 4h offset = 4006_2004h

Bit	31–0
R	ERDR
W	
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

ESAI_ERDR field descriptions

Field	Description
31–0 ERDR	ESAI Receive Data Register. Reading this register returns the data within the ESAI Receive FIFO. Reading this register when the Receive FIFO is empty returns the last valid data word. When multiple ESAI receivers are enabled, the data for each receiver is interleaved from lowest receiver to highest receiver (for example, if receivers 0, 2 and 3 are enabled then data is returned as follows: receiver #0, receiver #2, receiver #3, receiver #0, receiver #2, receiver #3, receiver #0, etc). Data is passed from the ESAI receive shift registers to the ESAI Receive FIFO as defined by the Receiver Word Alignment configuration bits either zero or sign-extended based on the Receive Extension control bit.

52.5.3 ESAI Control Register (ESAI_ECR)

Address: 4006_2000h base + 8h offset = 4006_2008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												ETI	ETO	ERI	ERO
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														ERST	ESAIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_ECR field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 ETI	EXTAL Transmitter In. Mux EXTAL in place of the High Frequency Transmitter Clock input pin. HCKT can still be used to drive a divided down EXTAL or as GPIO. 0 HCKT pin has normal function. 1 EXTAL muxed into HCKT input.
18 ETO	EXTAL Transmitter Out. Drive the EXTAL input on the High Frequency Transmitter Clock pin. 0 HCKT pin has normal function. 1 EXTAL driven onto HCKT pin.
17 ERI	EXTAL Receiver In. Mux EXTAL in place of the High Frequency Receiver Clock input pin. HCKR can still be used to drive a divided down EXTAL or as GPIO. 0 HCKR pin has normal function. 1 EXTAL muxed into HCKR input.
16 ERO	EXTAL Receiver Out. Drive the EXTAL input on the High Frequency Receiver Clock pin. 0 HCKR pin has normal function. 1 EXTAL driven onto HCKR pin.
15–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 ERST	ESAI Reset. Reset the ESAI core logic (including configuration registers) but not the ESAI FIFOs. 0 ESAI not reset. 1 ESAI reset.
0 ESAIEN	ESAI Enable. Enables/disables the ESAI logic clock. Enable the ESAI before reading or writing other ESAI registers.

Table continues on the next page...

ESAI_ECR field descriptions (continued)

Field	Description
	0 ESAI disabled. 1 ESAI enabled.

52.5.4 ESAI Status Register (ESAI_ESR)

Address: 4006_2000h base + Ch offset = 4006_200Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					TINIT	RFF	TFE	TLS	TDE	TED	TD	RLS	RDE	RED	RD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_ESR field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 TINIT	Transmit Initialization. Indicates that the Transmit FIFO is writing the first word for each enabled transmitter into the Transmit Data Registers. This bit sets when the Transmit FIFO is enabled (provided Transmit Initialization is enabled) and clears after the Transmit Data Registers have been initialized. The Transmit Enable bits in the Transmit Control Register should not be set until this flag has cleared. 0 Transmitter has finished initializing the Transmit Data Registers (or Transmit FIFO is not enabled or Transmit Initialization is not enabled). 1 Transmitter has not finished initializing the Transmit Data Registers.
9 RFF	Receive FIFO Full. Indicates that the number of data words in the Receive FIFO has equaled or exceeded the Receive FIFO Watermark. This flag also drives the ESAI Receiver DMA request line. ESAI FIFO DMA requests see ESAI DMA Requests from the FIFOs. 0 Number of words in Receive FIFO less than Receive FIFO watermark. 1 Number of words in Receive FIFO is equal to or greater than Receive FIFO watermark.
8 TFE	Transmit FIFO Empty. Indicates that the number of empty slots in the Transmit FIFO has met or exceeded the Transmit FIFO Watermark. This flag also drives the ESAI Transmitter DMA request line. ESAI FIFO DMA request see ESAI DMA Requests from the FIFOs. 0 Number of empty slots in Transmit FIFO less than Transmit FIFO watermark. 1 Number of empty slots in Transmit FIFO is equal to or greater than Transmit FIFO watermark.
7 TLS	Transmit Last Slot. Reading this register when TLS is set will negate the Transmit Last Slot interrupt. 0 TLS is not the highest priority active interrupt. 1 TLS is the highest priority active interrupt.

Table continues on the next page...

ESAI_ESR field descriptions (continued)

Field	Description
6 TDE	Transmit Data Exception. 0 TDE is not the highest priority active interrupt. 1 TDE is the highest priority active interrupt.
5 TED	Transmit Even Data. 0 TED is not the highest priority active interrupt. 1 TED is the highest priority active interrupt.
4 TD	Transmit Data. 0 TD is not the highest priority active interrupt. 1 TD is the highest priority active interrupt.
3 RLS	Receive Last Slot. Reading this register when RLS is set will negate the Receive Last Slot interrupt. 0 RLS is not the highest priority active interrupt. 1 RLS is the highest priority active interrupt.
2 RDE	Receive Data Exception. 0 RDE is not the highest priority active interrupt. 1 RDE is the highest priority active interrupt.
1 RED	Receive Even Data. 0 RED is not the highest priority active interrupt. 1 RED is the highest priority active interrupt.
0 RD	Receive Data. 0 RD is not the highest priority active interrupt. 1 RD is the highest priority active interrupt.

52.5.5 Transmit FIFO Configuration Register (ESAI_TFCR)

Address: 4006_2000h base + 10h offset = 4006_2010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												TIEN	TWA[2:0]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	TFWM[7:0]								TE5	TE4	TE3	TE2	TE1	TE0	TFR	TFE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TFCR field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ESAI_TFCR field descriptions (continued)

Field	Description
19 TIEN	Transmitter Initialization Enable. Enables the initialization of the Transmit Data Registers when the Transmitter FIFO is enabled. TIEN=1 is recommended. 0 Transmit Data Registers are not initialized from the FIFO once the Transmit FIFO is enabled. Software must manually initialize the Transmit Data Registers separately. 1 Transmit Data Registers are initialized from the FIFO once the Transmit FIFO is enabled.
18–16 TWA[2:0]	Transmit Word Alignment. Configures the alignment of the data written into the ESAI Transmit Data Register and then passed to the relevant 24 bit Transmit shift register. 000 MSB of data is bit 31. Data bits 7-0 are ignored when passed to transmit shift register. 001 MSB of data is bit 27. Data bits 3-0 are ignored when passed to transmit shift register. 010 MSB of data is bit 23. 011 MSB of data is bit 19. Bottom 4 bits of transmit shift register are zeroed. 100 MSB of data is bit 15. Bottom 8 bits of transmit shift register are zeroed. 101 MSB of data is bit 11. Bottom 12 bits of transmit shift register are zeroed. 110 MSB of data is bit 7. Bottom 16 bits of transmit shift register are zeroed. 111 MSB of data is bit 3. Bottom 20 bits of transmit shift register are zeroed.
15–8 TFWM[7:0]	Transmit FIFO Watermark. These bits configure the threshold at which the Transmit FIFO Empty flag will set. The TFE is set when the number of empty slots in the Transmit FIFO equal or exceed the selected threshold.
7 TE5	Transmitter #5 FIFO Enable. This bit enables transmitter #5 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #5 is not using the Transmit FIFO. 1 Transmitter #5 is using the Transmit FIFO.
6 TE4	Transmitter #4 FIFO Enable. This bit enables transmitter #4 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #4 is not using the Transmit FIFO. 1 Transmitter #4 is using the Transmit FIFO.
5 TE3	Transmitter #3 FIFO Enable. This bit enables transmitter #3 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #3 is not using the Transmit FIFO. 1 Transmitter #3 is using the Transmit FIFO.
4 TE2	Transmitter #2 FIFO Enable. This bit enables transmitter #2 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #2 is not using the Transmit FIFO. 1 Transmitter #2 is using the Transmit FIFO.
3 TE1	Transmitter #1 FIFO Enable. This bit enables transmitter #1 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #1 is not using the Transmit FIFO. 1 Transmitter #1 is using the Transmit FIFO.
2 TE0	Transmitter #0 FIFO Enable. This bit enables transmitter #0 to use the Transmit FIFO. Do not change this bit when the Transmitter FIFO is enabled. 0 Transmitter #0 is not using the Transmit FIFO. 1 Transmitter #0 is using the Transmit FIFO.

Table continues on the next page...

ESAI_TFCR field descriptions (continued)

Field	Description
1 TFR	Transmit FIFO Reset. This bit resets the Transmit FIFO pointers. 0 Transmit FIFO not reset. 1 Transmit FIFO reset.
0 TFE	Transmit FIFO Enable. This bit enables the use of the Transmit FIFO. 0 Transmit FIFO disabled. 1 Transmit FIFO enabled.

52.5.6 Transmit FIFO Status Register (ESAI_TFSR)

Address: 4006_2000h base + 14h offset = 4006_2014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	NTFO[2:0]			0	NTFI[2:0]			TFCNT[7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TFSR field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–12 NTFO[2:0]	Next Transmitter FIFO Out. Indicates which Transmit Data Register receives the top word of the Transmit FIFO. This will usually equal the lowest enabled transmitter, unless the transmit FIFO is empty. 000 Transmitter #0 receives next word from the Transmit FIFO. 001 Transmitter #1 receives next word from the Transmit FIFO. 010 Transmitter #2 receives next word from the Transmit FIFO. 011 Transmitter #3 receives next word from the Transmit FIFO. 100 Transmitter #4 receives next word from the Transmit FIFO. 101 Transmitter #5 receives next word from the Transmit FIFO. 110 Reserved. 111 Reserved.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 NTFI[2:0]	Next Transmitter FIFO In. Indicates which transmitter receives the next word written to the FIFO. 000 Transmitter #0 receives next word written to the Transmit FIFO. 001 Transmitter #1 receives next word written to the Transmit FIFO.

Table continues on the next page...

ESAI_TFSR field descriptions (continued)

Field	Description
	010 Transmitter #2 receives next word written to the Transmit FIFO. 011 Transmitter #3 receives next word written to the Transmit FIFO. 100 Transmitter #4 receives next word written to the Transmit FIFO. 101 Transmitter #5 receives next word written to the Transmit FIFO. 110 Reserved. 111 Reserved.
7–0 TFCNT[7:0]	Transmit FIFO Counter. These bits indicate the number of data words stored in the Transmit FIFO.

52.5.7 Receive FIFO Configuration Register (ESAI_RFCR)

Address: 4006_2000h base + 18h offset = 4006_2018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												REXT	RWA[2:0]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RFWM[7:0]								0		RE3	RE2	RE1	RE0	RFR	RFE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_RFCR field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 REXT	Receive Extension. Enables the receive data to be returned sign extended when the Receive Word Alignment is configured to return data where the MSB is not aligned with bit 31. 0 Receive data is zero extended. 1 Receive data is sign extended.
18–16 RWA[2:0]	Receive Word Alignment. Configures the alignment of the data passed from the relevant 24 bit Receive shift register and read out the ESAI Receive Data Register. 000 MSB of data is at bit 31. Data bits 7-0 are zeroed. 001 MSB of data is at bit 27. Data bits 3-0 are zeroed. 010 MSB of data is at bit 23. 011 MSB of data is at bit 19. Data bits 3-0 from receive shift register are ignored. 100 MSB of data is at bit 15. Data bits 7-0 from receive shift register are ignored. 101 MSB of data is at bit 11. Data bits 11-0 from receive shift register are ignored. 110 MSB of data is at bit 7. Data bits 15-0 from receive shift register are ignored. 111 MSB of data is at bit 3. Data bits 19-0 from receive shift register are ignored.

Table continues on the next page...

ESAI_RFCR field descriptions (continued)

Field	Description
15–8 RFWM[7:0]	Receive FIFO Watermark. These bits configure the threshold at which the Receive FIFO Full flag will set. The RFF is set when the number of words in the Receive FIFO equal or exceed the selected threshold. It can be set to a non-zero value.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 RE3	Receiver #3 FIFO Enable. This bit enables receiver #3 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #3 is not using the Receive FIFO. 1 Receiver #3 is using the Receive FIFO.
4 RE2	Receiver #2 FIFO Enable. This bit enables receiver #2 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #2 is not using the Receive FIFO. 1 Receiver #2 is using the Receive FIFO.
3 RE1	Receiver #1 FIFO Enable. This bit enables receiver #1 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #1 is not using the Receive FIFO. 1 Receiver #1 is using the Receive FIFO.
2 RE0	Receiver #0 FIFO Enable. This bit enables receiver #0 to use the Receive FIFO. Do not change this bit when the Receiver FIFO is enabled. 0 Receiver #0 is not using the Receive FIFO. 1 Receiver #0 is using the Receive FIFO.
1 RFR	Receive FIFO Reset. This bit resets the Receive FIFO pointers. 0 Receive FIFO not reset. 1 Receive FIFO reset.
0 RFE	Receive FIFO Enable. This bit enables the use of the Receive FIFO. 0 Receive FIFO disabled. 1 Receive FIFO enabled.

52.5.8 Receive FIFO Status Register (ESAI_RFSR)

Address: 4006_2000h base + 1Ch offset = 4006_201Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		NRFI[1:0]		0		NRFO[1:0]		RFCNT[7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_RFSR field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13–12 NRFI[1:0]	Next Receiver FIFO In. Indicates which Receiver Data Register the Receive FIFO will load next. This will usually equal the lowest enabled receiver, unless the receive FIFO is full. 00 Receiver #0 returns next word to the Receive FIFO. 01 Receiver #1 returns next word to the Receive FIFO. 10 Receiver #2 returns next word to the Receive FIFO. 11 Receiver #3 returns next word to the Receive FIFO.
11–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 NRFO[1:0]	Next Receiver FIFO Out. Indicates which receiver returns the top word of the Receive FIFO. 00 Receiver #0 returns next word from the Receive FIFO. 01 Receiver #1 returns next word from the Receive FIFO. 10 Receiver #2 returns next word from the Receive FIFO. 11 Receiver #3 returns next word from the Receive FIFO.
7–0 RFCNT[7:0]	Receive FIFO Counter. These bits indicate the number of data words stored in the Receive FIFO.

52.5.9 Transmit Data Register n (ESAI_TX*n*)

ESAI_TX5, ESAI_TX4, ESAI_TX3, ESAI_TX2, ESAI_TX1 and ESAI_TX0 are 32-bit write-only registers. Data to be transmitted is written into these registers and is automatically transferred to the transmit shift registers (Figure 52-2 and Figure 52-3). The data written (8, 12, 16, 20, or 24 bits) should occupy the most significant portion of the TXn according to the ALC control bit setting. The unused bits (least significant portion and the 8 most significant bits when ALC=1) of the TXn are don't care bits. The Core is interrupted whenever the TXn becomes empty if the transmit data register empty interrupt has been enabled.

Address: 4006_2000h base + 80h offset + (4d × i), where i=0d to 5d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0																							
W									TXn[23:0]																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TX*n* field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–0 TXn[23:0]	Stores the data to be transmitted and is automatically transferred to the transmit shift registers. See ESAI Transmit Shift Registers.

52.5.10 ESAI Transmit Slot Register (ESAI_TSR)

Address: 4006_2000h base + 98h offset = 4006_2098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0																							
W									TSR[23:0]																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TSR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–0 TSR[23:0]	The write-only Transmit Slot Register (ESAI_TSR) is effectively a null data register that is used when the data is not to be transmitted in the available transmit time slot. The transmit data pins of all the enabled transmitters are in the high-impedance state for the respective time slot where TSR has been written. The Transmitter External Buffer Enable pin (FSR pin when SYN=1, TEBE=1, RFSD=1) disables the external buffers during the slot when the ESAI_TSR register has been written.

52.5.11 Receive Data Register n (ESAI_RX*n*)

ESAI_RX3, ESAI_RX2, ESAI_RX1, and ESAI_RX0 are 32-bit read-only registers that accept data from the receive shift registers when they become full (Figure 52-2 and Figure 52-3). The data occupies the most significant portion of the receive data registers, according to the ALC control bit setting. The unused bits (least significant portion and 8 most significant bits when ALC=1) read as zeros. The Core is interrupted whenever RXn becomes full if the associated interrupt is enabled.

Address: 4006_2000h base + A0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RXn[23:0]																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_RX*n* field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–0 RXn[23:0]	Accept data from the receive shift registers when they become full See ESAI Receive Shift Registers

52.5.12 Serial Audio Interface Status Register (ESAI_SAISR)

The Status Register (ESAI_SAISR) is a read-only status register used by the ARM Core to read the status and serial input flags of the ESAI.

Address: 4006_2000h base + CCh offset = 4006_20CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														TODFE	TEDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TDE	TUE	TFS	0		RODF	REDF	RDF	ROE	RFS	0			IF2	IF1	IF0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_SAISR field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 TODFE	ESAI_SAISR Transmit Odd-Data Register Empty. When set, TODFE indicates that the enabled transmitter data registers became empty at the beginning of an odd time slot. Odd time slots are all odd-numbered slots (1, 3, 5, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TODFE indicates that data should be written to all the TX registers of the enabled transmitters or to the transmit slot register (ESAI_TSR). TODE is cleared when the Core writes to all the transmit data registers of the enabled transmitters, or when the Core writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TODFE is set. Hardware, software, ESAI individual reset clear TODFE.
16 TEDE	ESAI_SAISR Transmit Even-DataRegister Empty. When set, TEDE indicates that the enabled transmitter data registers became empty at the beginning of an even time slot. Even time slots are all even-numbered slots (0, 2, 4, 6, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TEDE indicates that data should be written to all the TX registers of the enabled transmitters or to the transmit slot register (ESAI_TSR). TEDE is cleared when the Core writes to all the transmit data registers of the enabled transmitters, or when the Core writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TEDE is set. Hardware, software, ESAI individual reset clear TEDE.
15 TDE	ESAI_SAISR Transmit Data Register Empty. TDE is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TDE indicates that data should be written to all the TX registers of the enabled transmitters or to the transmit slot register (ESAI_TSR). TDE is cleared when the Core writes to all the transmit data registers of the enabled transmitters, or when the Core writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TDE is set. Hardware, software, ESAI individual reset clear TDE.
14 TUE	ESAI_SAISR Transmit Underrun Error Flag. TUE is set when at least one of the enabled serial transmit shift registers is empty (no new data to be transmitted) and a transmit time slot occurs. When a transmit underrun error occurs, the previous data (which is still present in the TX registers that were not written) is retransmitted. If TEIE is set, an ESAI transmit data with exception (underrun error) interrupt request is issued when TUE is set. Hardware, software, ESAI individual reset clear TUE. TUE is also cleared by reading the ESAI_SAISR with TUE set, followed by writing to all the enabled transmit data registers or to ESAI_TSR.
13 TFS	ESAI_SAISR Transmit Frame Sync Flag. When set, TFS indicates that a transmit frame sync occurred in the current time slot. TFS is set at the start of the first time slot in the frame and cleared during all other time slots. Data written to a transmit data register during the time slot when TFS is set is transmitted (in network mode), if the transmitter is enabled, during the second time slot in the frame. TFS is useful in network mode to identify the start of a frame. TFS is cleared by hardware, software, ESAI individual reset. TFS is valid only if at least one transmitter is enabled, that is, one or more of TE0, TE1, TE2, TE3, TE4 and TE5 are set. (In normal mode, TFS always reads as a one when transmitting data because there is only one time slot per frame - the "frame sync" time slot)
12–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 RODF	ESAI_SAISR Receive Odd-Data Register Full. When set, RODF indicates that the received data in the receive data registers of the enabled receivers have arrived during an odd time slot when operating in the network mode. Odd time slots are all odd-numbered slots (1, 3, 5, and so on). Time slots are numbered

Table continues on the next page...

ESAI_SAISR field descriptions (continued)

Field	Description
	from zero to N-1, where N is the number of time slots in the frame. RODF is set when the contents of the receive shift registers are transferred to the receive data registers. RODF is cleared when the Core reads all the enabled receive data registers or cleared by hardware, software, ESAI individual resets.
9 REDF	ESAI_SAISR Receive Even-Data Register Full. When set, REDF indicates that the received data in the receive data registers of the enabled receivers have arrived during an even time slot when operating in the network mode. Even time slots are all even-numbered slots (0, 2, 4, 6, and so on). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. REDF is set when the contents of the receive shift registers are transferred to the receive data registers. REDF is cleared when the Core reads all the enabled receive data registers or cleared by hardware, software, ESAI individual resets. If REDIE is set, an ESAI receive even slot data interrupt request is issued when REDF is set.
8 RDF	ESAI_SAISR Receive Data Register Full. RDF is set when the contents of the receive shift register of an enabled receiver is transferred to the respective receive data register. RDF is cleared when the Core reads the receive data register of all enabled receivers or cleared by hardware, software, ESAI individual reset. If RIE is set, an ESAI receive data interrupt request is issued when RDF is set.
7 ROE	ESAI_SAISR Receive Overrun Error Flag. The ROE flag is set when the serial receive shift register of an enabled receiver is full and ready to transfer to its receiver data register (RXn) and the register is already full (RDF=1). If REIE is set, an ESAI receive data with exception (overrun error) interrupt request is issued when ROE is set. Hardware, software, ESAI individual reset clear ROE. ROE is also cleared by reading the SAISR with ROE set, followed by reading all the enabled receive data registers.
6 RFS	ESAI_SAISR Receive Frame Sync Flag. When set, RFS indicates that a receive frame sync occurred during reception of the words in the receiver data registers. This indicates that the data words are from the first slot in the frame. When RFS is clear and a word is received, it indicates (only in the network mode) that the frame sync did not occur during reception of that word. RFS is cleared by hardware, software, ESAI individual reset. RFS is valid only if at least one of the receivers is enabled (REx=1). (In normal mode, RFS always reads as a one when reading data because there is only one time slot per frame - the "frame sync" time slot)
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 IF2	ESAI_SAISR Serial Input Flag 2. The IF2 bit is enabled only when the HCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RHCKD=0, indicating that HCKR is an input flag and the synchronous mode is selected. Data present on the HCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF2 bit is updated with this data when the receive shift registers are transferred into the receiver data registers. IF2 reads as a zero when it is not enabled. Hardware, software, ESAI individual reset clear IF2.
1 IF1	ESAI_SAISR Serial Inout Flag 1. The IF1 bit is enabled only when the FSR pin is defined as ESAI in the Port Control Register, SYN =1, RFSD=0 and TEBE=0, indicating that FSR is an input flag and the synchronous mode is selected. Data present on the FSR pin is latched during reception of the first received data bit after frame sync is detected. The IF1 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF1 reads as a zero when it is not enabled. Hardware, software, ESAI individual reset clear IF1.
0 IF0	ESAI_SAISR Serial Input Flag 0. The IF0 bit is enabled only when the SCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RCKD=0, indicating that SCKR is an input flag and the synchronous mode is selected. Data present on the SCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF0 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF0 reads as a zero when it is not enabled. Hardware, software, ESAI individual reset clear IF0.

52.5.13 Serial Audio Interface Control Register (ESAI_SAICR)

The read/write Common Control Register (ESAI_SAICR) contains control bits for functions that affect both the receive and transmit sections of the ESAI.

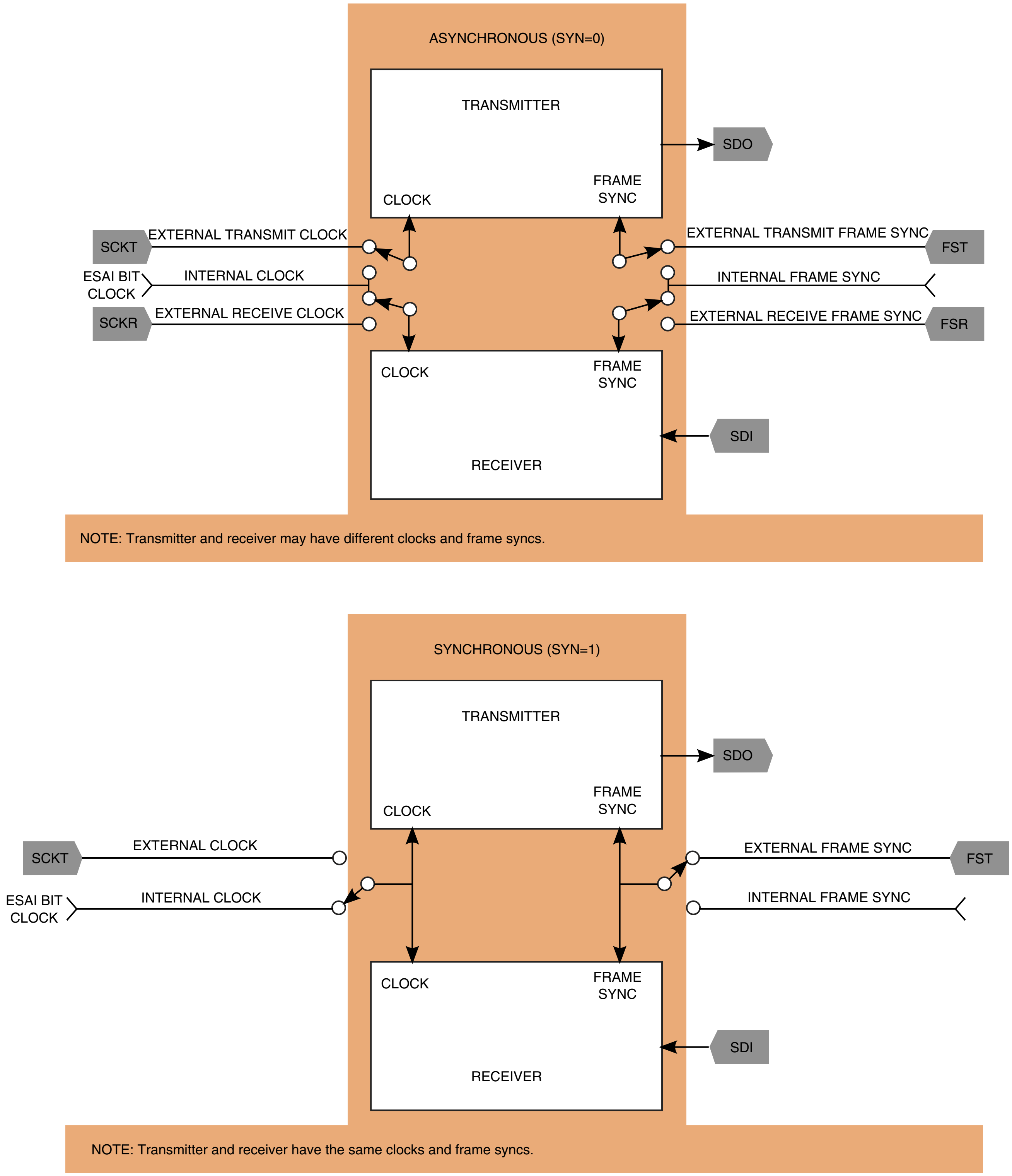


Figure 52-27. SAICR SYN Bit Operation

Address: 4006_2000h base + D0h offset = 4006_20D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							ALC	TEBE	SYN	0			OF2	OF1	OF0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_SAICR field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 ALC	ESAI_SAICR Alignment Control. The ESAI is designed for 24-bit fractional data, thus shorter data words are left aligned to the MSB (bit 23). Some applications use 16-bit fractional data. In those cases, shorter data words may be left aligned to bit 15. The Alignment Control (ALC) bit supports these applications. If ALC is set, transmitted and received words are left aligned to bit 15 in the transmit and receive shift registers. If ALC is cleared, transmitted and received word are left aligned to bit 23 in the transmit and receive shift registers. While ALC is set, 20-bit and 24-bit words may not be used, and word length control should specify 8-, 12-, or 16-bit words; otherwise, results are unpredictable.
7 TEBE	ESAI_SAICR Transmit External Buffer Enable. The Transmitter External Buffer Enable (TEBE) bit controls the function of the FSR pin when in the synchronous mode. If the ESAI is configured for operation in the synchronous mode (SYN=1), and TEBE is set while FSR pin is configured as an output (RFSD=1), the FSR pin functions as the transmitter external buffer enable control to enable the use of an external buffers on the transmitter outputs. If TEBE is cleared, the FSR pin functions as the serial I/O flag 1. See Port C Control Register for a summary of the effects of TEBE on the FSR pin.
6 SYN	ESAI_SAICR Synchronous Mode Selection. The Synchronous Mode Selection (SYN) bit controls whether the receiver and transmitter sections of the ESAI operate synchronously or asynchronously with respect to each other (see Port C Control Register). When SYN is cleared, the asynchronous mode is chosen and independent clock and frame sync signals are used for the transmit and receive sections. When SYN is set, the synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals. When in the synchronous mode (SYN=1), the transmit and receive sections use the transmitter section clock generator as the source of the clock and frame sync for both sections. Also, the receiver clock pins SCKR, FSR and HCKR now operate as I/O flags. Refer to Table 52-38, Table 52-39, and Table 52-40 for the effects of SYN on the receiver clock pins.
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 OF2	ESAI_SAICR Serial Output Flag 2. The Serial Output Flag 2 (OF2) is a data bit used to hold data to be send to the OF2 pin. When the ESAI is in the synchronous clock mode (SYN=1), the HCKR pin is configured as the ESAI flag 2. If the receiver high frequency clock direction bit (RHCKD) is set, the HCKR pin is the output flag OF2, and data present in the OF2 bit is written to the OF2 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.
1 OF1	ESAI_SAICR Serial Output Flag 1. The Serial Output Flag 1 (OF1) is a data bit used to hold data to be send to the OF1 pin. When the ESAI is in the synchronous clock mode (SYN=1), the FSR pin is configured as the ESAI flag 1. If the receiver frame sync direction bit (RFSD) is set and the TEBE bit is cleared, the FSR pin is the output flag OF1, and data present in the OF1 bit is written to the OF1 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

Table continues on the next page...

ESAI_SAICR field descriptions (continued)

Field	Description
0 OF0	ESAI_SAICR Serial Output Flag 0. The Serial Output Flag 0 (OF0) is a data bit used to hold data to be send to the OF0 pin. When the ESAI is in the synchronous clock mode (SYN=1), the SCKR pin is configured as the ESAI flag 0. If the receiver serial clock direction bit (RCKD) is set, the SCKR pin is the output flag OF0, and data present in the OF0 bit is written to the OF0 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

52.5.14 Transmit Control Register (ESAI_TCR)

The read/write Transmit Control Register (ESAI_TCR) controls the ESAI transmitter section. Interrupt enable bits for the transmitter section are provided in this control register. Operating modes are also selected in this register.

Table 52-29. Transmit Network Mode Selection

TMOD1	TMOD0	TDC4-TDC0	Transmitter Network Mode
0	0	0x0-0x1F	Normal Mode
0	1	0x0	On-Demand Mode
0	1	0x1-0x1F	Network Mode
1	0	X	Reserved
1	1	0x0C	AC97

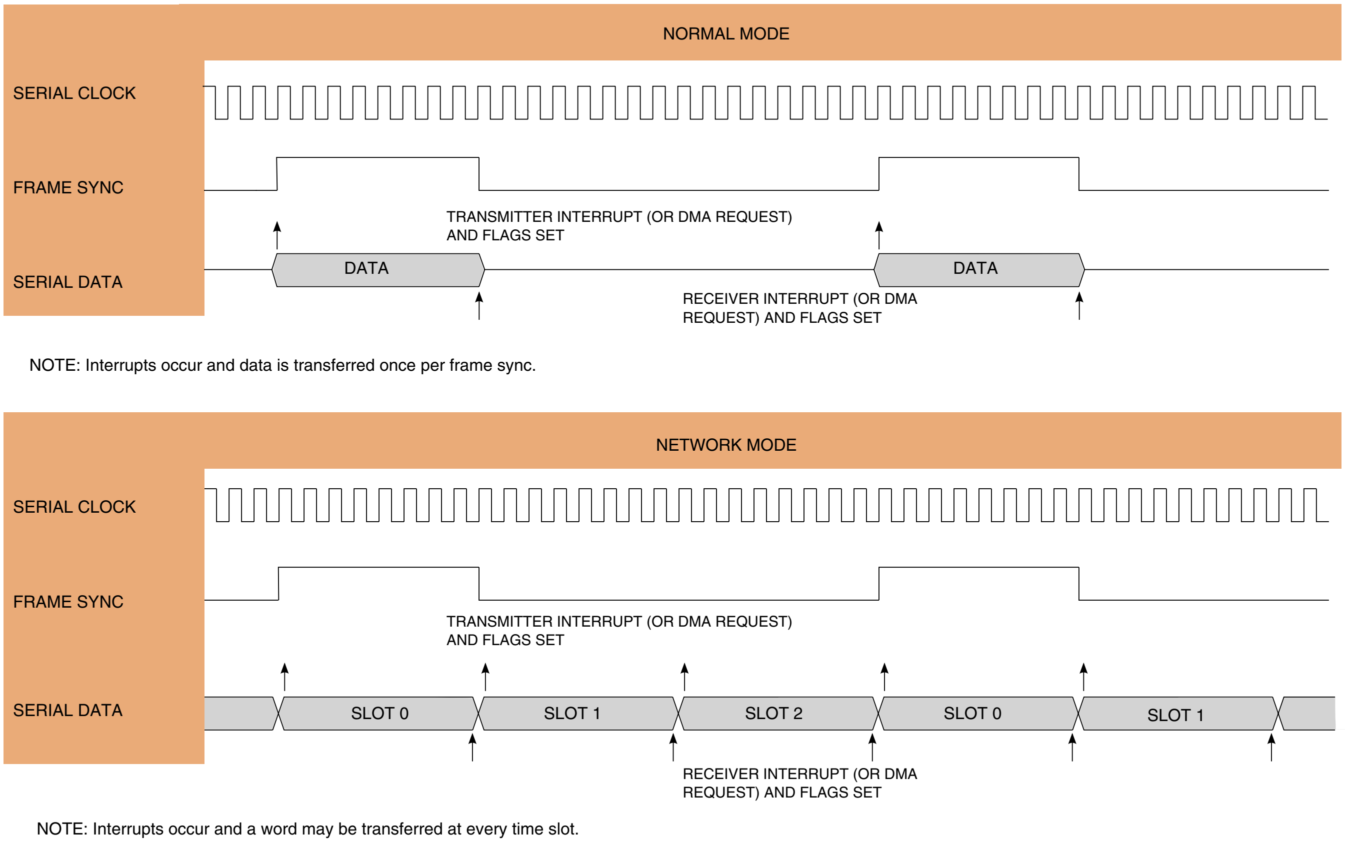


Figure 52-29. Normal and Network Operation

Table 52-30. ESAI Transmit Slot and Word Length Selection

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20

Table continues on the next page...

Table 52-30. ESAI Transmit Slot and Word Length Selection (continued)

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

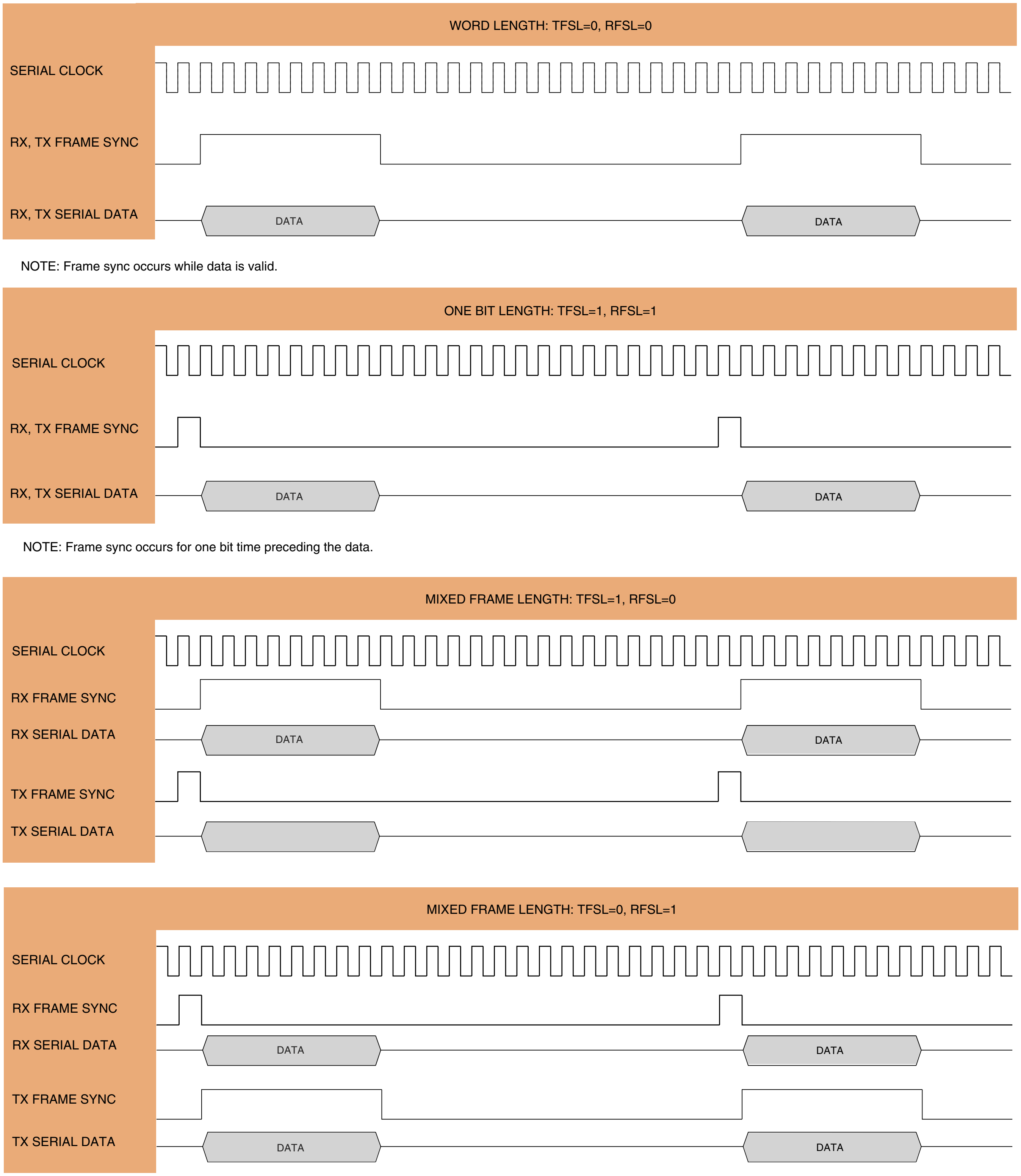


Figure 52-30. Frame Length Selection

Address: 4006_2000h base + D4h offset = 4006_20D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								TLIE	TIE	TEDIE	TEIE	TPR	0	PADC	TFSR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	TFSL	TSWS[14:10]					TMOD[9:8]		TWA	TSHFD	TE5	TE4	TE3	TE2	TE1	TE0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TCR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 TLIE	ESAI_TCR Transmit Last Slot Interrupt Enable. TLIE enables an interrupt at the beginning of last slot of a frame in network mode. When TLIE is set the Core is interrupted at the start of the last slot in a frame in network mode regardless of the transmit mask register setting. When TLIE is cleared the transmit last slot interrupt is disabled. TLIE is disabled when TDC[4:0]=0x00000 (on-demand mode). The use of the transmit last slot interrupt is described in ESAI Interrupt Requests.
22 TIE	ESAI_TCR Transmit Interrupt Enable. The Core is interrupted when TIE and the TDE flag in the ESAI_SAISR status register are set. When TIE is cleared, this interrupt is disabled. Writing data to all the data registers of the enabled transmitters or to ESAI_TSR clears TDE, thus clearing the interrupt. Transmit interrupts with exception have higher priority than normal transmit data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.
21 TEDIE	ESAI_TCR Transmit Even Slot Data Interrupt Enable. The TEDIE control bit is used to enable the transmit even slot data interrupts. If TEDIE is set, the transmit even slot data interrupts are enabled. If TEDIE is cleared, the transmit even slot data interrupts are disabled. A transmit even slot data interrupt request is generated if TEDIE is set and the TEDE status flag in the ESAI_SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot in the frame is marked by the frame sync signal and is considered to be even. Writing data to all the data registers of the enabled transmitters or to ESAI_TSR clears the TEDE flag, thus servicing the interrupt. Transmit interrupts with exception have higher priority than transmit even slot data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.
20 TEIE	ESAI_TCR Transmit Exception Interrupt Enable. When TEIE is set, the Core is interrupted when both TDE and TUE in the ESAI_SAISR status register are set. When TEIE is cleared, this interrupt is disabled. Reading the ESAI_SAISR status register followed by writing to all the data registers of the enabled transmitters clears TUE, thus clearing the pending interrupt.
19 TPR	ESAI_TCR Transmit Section Personal Reset. The TPR control bit is used to put the transmitter section of the ESAI in the personal reset state. The receiver section is not affected. When TPR is cleared, the transmitter section may operate normally. When TPR is set, the transmitter section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset. The control bits are not affected by the personal reset state. The transmitter data pins are tri-stated while in the personal reset state; if a stable logic level is desired, the transmitter data pins should be defined as GPIO outputs, or external pull-up or pull-down resistors should be used. The

Table continues on the next page...

ESAI_TCR field descriptions (continued)

Field	Description
	transmitter clock outputs drive zeroes while in the personal reset state. Note that to leave the personal reset state by clearing TPR, the procedure described in ESAI Initialization Examples should be followed.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 PADC	ESAI_TCR Transmit Zero Padding Control. When PADC is cleared, zero padding is disabled. When PADC is set, zero padding is enabled. PADC, in conjunction with the TWA control bit, determines the way that padding is done for operating modes where the word length is less than the slot length. See the TWA bit description in bit 7 for more details. Because the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule: If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted. If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.
16 TFSR	ESAI_TCR Transmit Frame Sync Relative Timing. TFSR determines the relative timing of the transmit frame sync signal as referred to the serial data lines, for a word length frame sync only (TFSL=0). When TFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When TFSR is set the word length frame sync starts one serial clock cycle earlier, that is, together with the last bit of the previous data word.
15 TFSL	ESAI_TCR Transmit Frame Sync Length. The TFSL bit selects the length of frame sync to be generated or recognized. If TFSL is cleared, a word-length frame sync is selected. If TFSL is set, a 1-bit clock period frame sync is selected. See Figure 1-21 for examples of frame length selection.
14–10 TSWS[14:10]	ESAI_TCR Tx Slot and Word Length Select (TSWS4-TSWS0). The TSWS4-TSWS0 bits are used to select the length of the slot and the length of the data words being transferred through the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 52-30. See also the ESAI data path programming model in Figure 52-2 and Figure 52-3.
9–8 TMOD[9:8]	ESAI_TCR Transmit Network Mode Control (TMOD1-TMOD0). The TMOD1 and TMOD0 bits are used to define the network mode of ESAI transmitters, as shown in Table 52-30. In the normal mode, the frame rate divider determines the word transfer rate - one word is transferred per frame sync during the frame sync time slot, as shown in Figure 52-29. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 52-29. For further details, refer to Modes of Operation In order to comply with AC-97 specifications, TSWS4-TSWS0 should be set to 00011 (20-bit slot, 20-bit word length), TFSL and TFSR should be cleared, and TDC4-TDC0 should be set to 0x0C (13 words in frame). If TMOD[1:0]=0b11 and the above recommendations are followed, the first slot and word will be 16 bits long, and the next 12 slots and words will be 20 bits long, as required by the AC97 protocol.
7 TWA	ESAI_TCR Transmit Word Alignment Control. The Transmitter Word Alignment Control (TWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If TWA is cleared, the data word is left-aligned in the slot frame during transmission. If TWA is set, the data word is right-aligned in the slot frame during transmission. Because the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule: If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted. If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.

Table continues on the next page...

ESAI_TCR field descriptions (continued)

Field	Description
6 TSHFD	ESAI_TCR Transmit Shift Direction. The TSHFD bit causes the transmit shift registers to shift data out MSB first when TSHFD equals zero or LSB first when TSHFD equals one (see Figure 52-2 and Figure 52-3 .
5 TE5	ESAI_TCR ESAI Transmit 5 Enable. TE5 enables the transfer of data from ESAI_TX5 to the transmit shift register #5. When TE5 is set and a frame sync is detected, the transmit #5 portion of the ESAI is enabled for that frame. When TE5 is cleared, the transmitter #5 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX5 when TE5 is cleared but the data is not transferred to the transmit shift register #5. The SDO5/SDI0 pin is the data input pin for ESAI_RX0 if TE5 is cleared and RE0 in the ESAI_RCR register is set. If both RE0 and TE5 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE0 and TE5 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE5 and setting it again disables the transmitter #5 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO5/SDI0 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE5 can be left enabled.
4 TE4	ESAI_TCR ESAI Transmit 4 Enable. TE4 enables the transfer of data from ESAI_TX4 to the transmit shift register #4. When TE4 is set and a frame sync is detected, the transmit #4 portion of the ESAI is enabled for that frame. When TE4 is cleared, the transmitter #4 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX4 when TE4 is cleared but the data is not transferred to the transmit shift register #4. The SDO4/SDI1 pin is the data input pin for ESAI_RX1 if TE4 is cleared and RE1 in the RCR register is set. If both RE1 and TE4 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE1 and TE4 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE4 and setting it again disables the transmitter #4 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO4/SDI1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE4 can be left enabled.
3 TE3	ESAI_TCR ESAI Transmit 3 Enable. TE3 enables the transfer of data from ESAI_TX3 to the transmit shift register #3. When TE3 is set and a frame sync is detected, the transmit #3 portion of the ESAI is enabled for that frame. When TE3 is cleared, the transmitter #3 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to ESAI_TX3 when TE3 is cleared but the data is not transferred to the transmit shift register #3. The SDO3/SDI2 pin is the data input pin for ESAI_RX2 if TE3 is cleared and RE2 in the ESAI_RCR register is set. If both RE2 and TE3 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE2 and TE3 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE3 and setting it again disables the transmitter #3 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO3/SDI2 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE3 can be left enabled.
2 TE2	ESAI_TCR ESAI Transmit 2 Enable. TE2 enables the transfer of data from ESAI_TX2 to the transmit shift register #2. When TE2 is set and a frame sync is detected, the transmit #2 portion of the ESAI is enabled for that frame. When TE2 is cleared, the transmitter #2 is disabled after completing transmission of data

Table continues on the next page...

ESAI_TCR field descriptions (continued)

Field	Description
	currently in the ESAI transmit shift register. Data can be written to ESAI_TX2 when TE2 is cleared but the data is not transferred to the transmit shift register #2. The SDO2/SDI3 pin is the data input pin for ESAI_RX3 if TE2 is cleared and RE3 in the ESAI_RCR register is set. If both RE3 and TE2 are cleared, the transmitter and receiver are disabled, and the pin is tri-stated. Both RE3 and TE2 should not be set at the same time. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE2 and setting it again disables the transmitter #2 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO2/SDI3 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE2 can be left enabled.
1 TE1	ESAI_TCR ESAI Transmit 1 Enable. TE1 enables the transfer of data from TX1 to the transmit shift register #1. When TE1 is set and a frame sync is detected, the transmit #1 portion of the ESAI is enabled for that frame. When TE1 is cleared, the transmitter #1 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO1 output is tri-stated, and any data present in TX1 is not transmitted, that is, data can be written to TX1 with TE1 cleared, but data is not transferred to the transmit shift register #1. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE1 and setting it again disables the transmitter #1 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE1 can be left enabled.
0 TE0	ESAI_TCR ESAI Transmit 0 Enable. TE0 enables the transfer of data from ESAI_TX0 to the transmit shift register #0. When TE0 is set and a frame sync is detected, the transmit #0 portion of the ESAI is enabled for that frame. When TE0 is cleared, the transmitter #0 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO0 output is tri-stated, and any data present in ESAI_TX0 is not transmitted, that is, data can be written to ESAI_TX0 with TE0 cleared, but data is not transferred to the transmit shift register #0. The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one. In the network mode, the operation of clearing TE0 and setting it again disables the transmitter #0 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO0 pin remains in the high-impedance state.The on-demand mode transmit enable sequence can be the same as the normal mode, or TE0 can be left enabled.

52.5.15 Transmit Clock Control Register (ESAI_TCCR)

The read/write Transmitter Clock Control Register (ESAI_TCCR) controls the ESAI transmitter clock generator bit and frame sync rates, the bit clock and high frequency clock sources and the directions of the HCKT, FST and SCKT signals. In the synchronous mode (SYN=1), the bit clock defined for the transmitter determines the receiver bit clock as well. ESAI_TCCR also controls the number of words per frame for the serial data. Hardware and software reset clear all the bits of the ESAI_TCCR register.

Care should be taken in asynchronous mode whenever the frame sync clock (FSR, FST) is not sourced directly from its associated bit clock (SCKR, SCKT). Proper phase relationships must be maintained between these clocks in order to guarantee proper operation of the ESAI.

NOTE

ARM Core clock is ipg_clk_esai in block ESAI which is from CCM's ahb_clk_root.

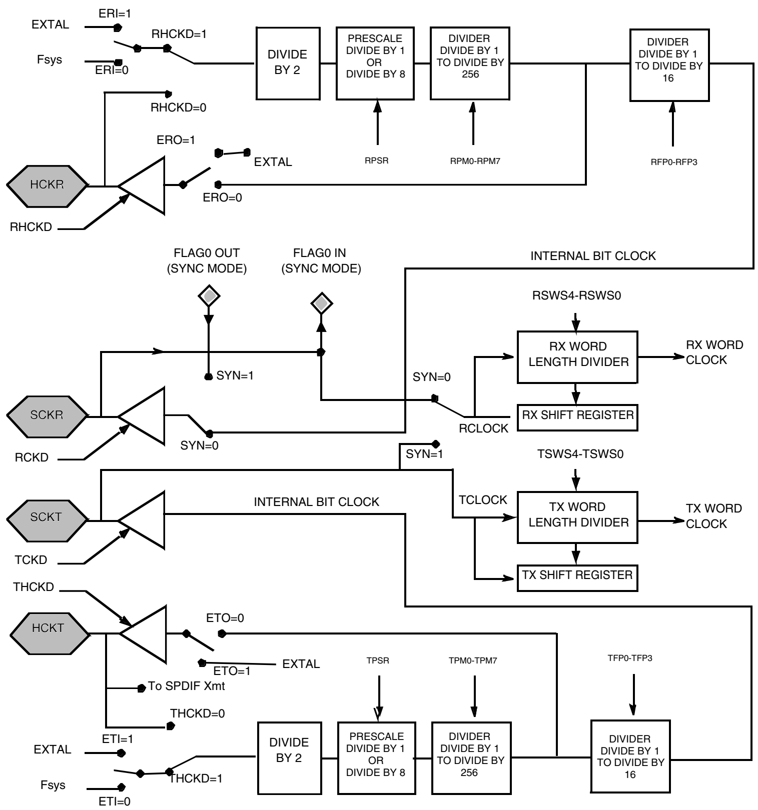


Figure 52-32. ESAI Clock Generator Functional Block Diagram

NOTE

1. ETI, ETO, ERI and ERO bit descriptions are covered in ESAI Control Register (ESAI_ECR). 2. Fsys is the ESAI system 133 MHz clock. 3. EXTAL is the on-chip clock sources other than ESAI system 133MHz clock.

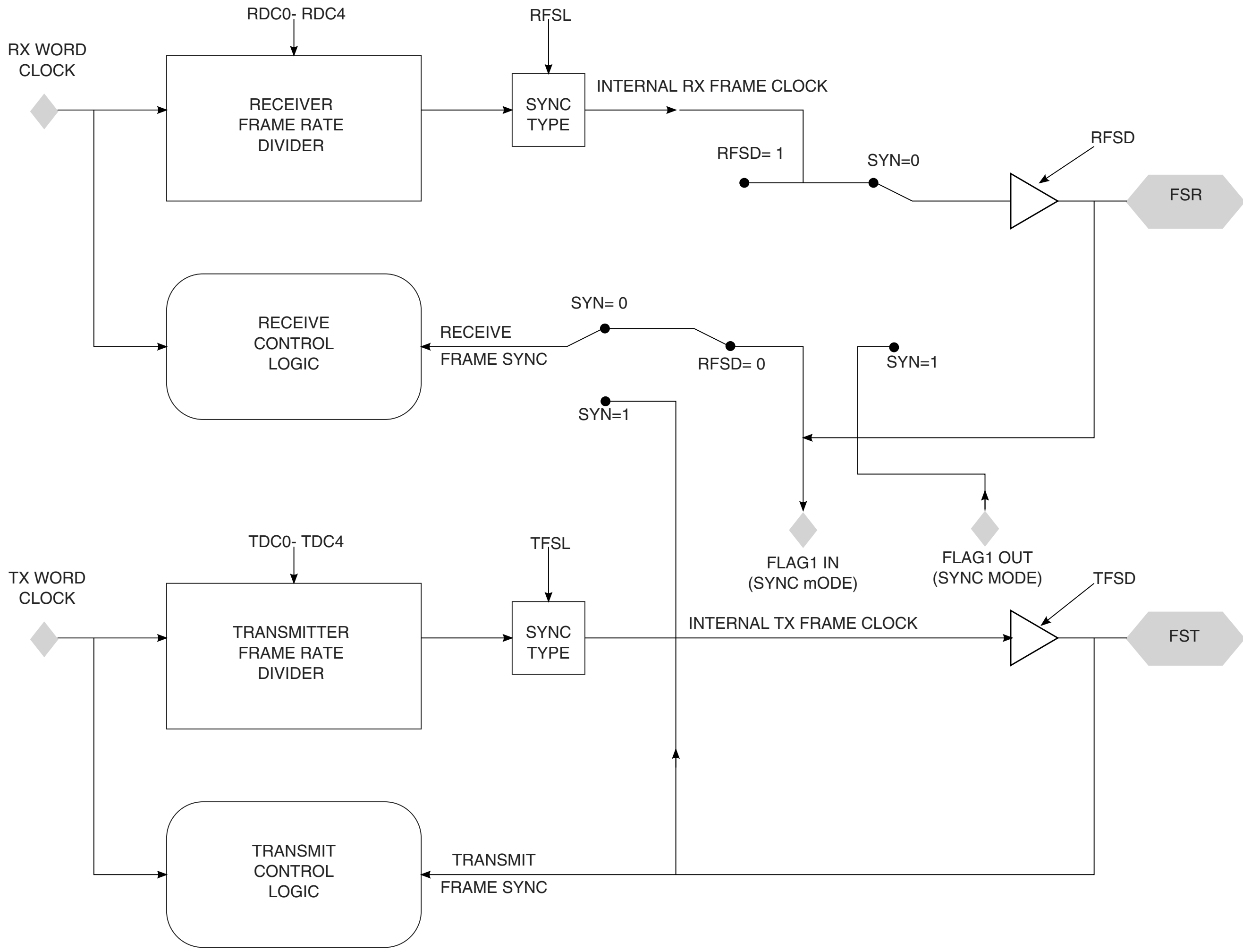


Figure 52-33. ESAI Frame Sync Generator Functional Block Diagram

Table 52-32. Transmitter High Frequency Clock Divider

TFP3-TFP0	Divide Ratio
0x0	1
0x1	2
0x2	3
0x3	4
...	...
0xF	16

Address: 4006_2000h base + D8h offset = 4006_20D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								THCKD	TFSD	TCKD	THCKP	TFSP	TCKP	TFP[3:0]	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	TFP[3:0]		TDC[4:0]					TPSR	TPM[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_TCCR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 THCKD	ESAI_TCCR Transmit High Frequency Clock Direction. THCKD controls the direction of the HCKT pin. When THCKD is cleared, HCKT is an input; when THCKD is set, HCKT is an output (see Table 52-3).
22 TFSD	ESAI_TCCR Transmit Frame Sync Signal Direction. TFSD controls the direction of the FST pin. When TFSD is cleared, FST is an input; when TFSD is set, FST is an output (see Table 52-3).
21 TCKD	ESAI_TCCR Transmit Clock Source Direction. The Transmitter Clock Source Direction (TCKD) bit selects the source of the clock signal used to clock the transmit shift registers in the asynchronous mode (SYN=0) and the transmit shift registers and the receive shift registers in the synchronous mode (SYN=1). When TCKD is set, the internal clock source becomes the bit clock for the transmit shift registers and word length divider and is the output on the SCKT pin. When TCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKT pin, and an external clock source may drive this pin (see Table 52-3).
20 THCKP	ESAI_TCCR Transmit High Frequency Clock Polarity. The Transmitter High Frequency Clock Polarity (THCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If THCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit high frequency bit clock and latched in on the falling edge of the transmit bit clock. If THCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.
19 TFSP	ESAI_TCCR Transmit Frame Sync Polarity. The Transmitter Frame Sync Polarity (TFSP) bit determines the polarity of the transmit frame sync signal. When TFSP is cleared, the frame sync signal polarity is positive, that is, the frame start is indicated by a high level on the frame sync pin. When TFSP is set, the frame sync signal polarity is negative, that is, the frame start is indicated by a low level on the frame sync pin.
18 TCKP	ESAI_TCCR Transmit Clock Polarity. The Transmitter Clock Polarity (TCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If TCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit bit clock and latched in on the falling edge of the transmit bit clock. If TCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.
17–14 TFP[3:0]	ESAI_TCCR Tx High Frequency Clock Divider. The TFP3-TFP0 bits control the divide ratio of the transmitter high frequency clock to the transmitter serial bit clock when the source of the high frequency clock and the bit clock is the internal ARM Core clock. When the HCKT input is being driven from an external high frequency clock, the TFP3-TFP0 bits specify an additional division ratio in the clock divider

Table continues on the next page...

ESAI_TCCR field descriptions (continued)

Field	Description
	chain. Table 52-32 shows the specification for the divide ratio. Figure 52-32 shows the ESAI high frequency clock generator functional diagram.
13–9 TDC[4:0]	ESAI_TCCR Tx Frame Rate Divider Control. The TDC4-TDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the transmitter frame clocks. In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (TDC[4:0]=0x00001 to 0x11111) for network mode. A divide ratio of one (TDC[4:0]=0x00000) in network mode is a special case (on-demand mode). In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (TDC[4:0]=0x00000 to 0x11111) for normal mode. In normal mode, a divide ratio of 1 (TDC[4:0]=0x00000) provides continuous periodic data word transfers. A bit-length frame sync (TFSL=1) must be used in this case. The ESAI frame sync generator functional diagram is shown in Figure 52-33
8 TPSR	ESAI_TCCR Transmit Prescaler Range. The TPSR bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When TPSR is set, the fixed prescaler is bypassed. When TPSR is cleared, the fixed divide-by-eight prescaler is operational (see Figure 52-32). The maximum internally generated bit clock frequency is Fsys/4; the minimum internally generated bit clock frequency is Fsys/(2 x 8 x 256 x 16)=Fsys/65536. (Do not use the combination TPSR=1, TPM7-TPM0=0x00, and TFP3-TFP0=0x0 which causes synchronization problems when using the internal ARM Core clock as source (TCKD=1 or THCKD=1))
7–0 TPM[7:0]	ESAI_TCCR Transmit Prescale Modulus Select. The TPM7-TPM0 bits specify the divide ratio of the prescale divider in the ESAI transmitter clock generator. A divide ratio from 1 to 256 (TPM[7:0]=0x00 to 0xFF) may be selected. The bit clock output is available at the transmit serial bit clock (SCKT) pin. The bit clock output is also available internally for use as the bit clock to shift the transmit and receive shift registers. The ESAI transmit clock generator functional diagram is shown in Figure 52-32.

52.5.16 Receive Control Register (ESAI_RCR)

The read/write Receive Control Register (ESAI_RCR) controls the ESAI receiver section. Interrupt enable bits for the receivers are provided in this control register. The receivers are enabled in this register (0,1,2 or 3 receivers can be enabled) if the input data pin is not used by a transmitter. Operating modes are also selected in this register.

Table 52-34. ESAI Receive Network Mode Selection

RMOD1	RMOD0	RDC4-RDC0	Receiver Network Mode
0	0	0x0-0x1F	Normal Mode
0	1	0x0	On-Demand Mode
0	1	0x1-0x1F	Network Mode
1	0	X	Reserved
1	1	0x0C	AC97

Table 52-35. ESAI Receive Slot and Word Length Selection

RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

Address: 4006_2000h base + DCh offset = 4006_20DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								RLIE	RIE	REDIE	REIE	RPR	0		RFSR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RFSL	RSWS[4:0]					RMOD[1:0]		RWA	RSHFD	0		RE3	RE2	RE1	RE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_RCR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 RLIE	ESAI_RCR Receive Last Slot Interrupt Enable. RLIE enables an interrupt after the last slot of a frame ended in network mode only. When RLIE is set the Core is interrupted after the last slot in a frame ended regardless of the receive mask register setting. When RLIE is cleared the receive last slot interrupt is disabled. Hardware and software reset clear RLIE. RLIE is disabled when RDC[4:0]=00000 (on-demand mode). The use of the receive last slot interrupt is described in ESAI Interrupt Requests.
22 RIE	ESAI_RCR Receive Interrupt Enable. The Core is interrupted when RIE and the RDF flag in the ESAI_SAISR status register are set. When RIE is cleared, this interrupt is disabled. Reading the receive data registers of the enabled receivers clears RDF, thus clearing the interrupt. Receive interrupts with exception have higher priority than normal receive data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.
21 REDIE	ESAI_RCR Receive Even Slot Data Interrupt Enable. The REDIE control bit is used to enable the receive even slot data interrupts. If REDIE is set, the receive even slot data interrupts are enabled. If REDIE is cleared, the receive even slot data interrupts are disabled. A receive even slot data interrupt request is generated if REDIE is set and the REDF status flag in the ESAI_SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot is marked by the frame sync signal and is considered to be even. Reading all the data registers of the enabled receivers clears the REDF flag, thus servicing the interrupt. Receive interrupts with exception have higher priority than receive even slot data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.
20 REIE	ESAI_RCR Receive Exception Interrupt Enable. When REIE is set, the Core is interrupted when both RDF and ROE in the ESAI_SAISR status register are set. When REIE is cleared, this interrupt is disabled. Reading the ESAI_SAISR status register followed by reading the enabled receivers data registers clears ROE, thus clearing the pending interrupt.
19 RPR	ESAI_RCR Receiver Section Personal Reset. The RPR control bit is used to put the receiver section of the ESAI in the personal reset state. The transmitter section is not affected. When RPR is cleared, the receiver section may operate normally. When RPR is set, the receiver section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset.The control bits are not affected by the personal reset state.The receiver data pins are disconnected while in the personal reset state.

Table continues on the next page...

ESAI_RCR field descriptions (continued)

Field	Description
	NOTE: To leave the personal reset state by clearing RPR, the procedure described in ESAI Initialization Examples should be followed.
18–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 RFSR	ESAI_RCR Receiver Frame Sync Relative Timing. RFSR determines the relative timing of the receive frame sync signal as referred to the serial data lines, for a word length frame sync only. When RFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When RFSR is set the word length frame sync starts one serial clock cycle earlier, that is, together with the last bit of the previous data word.
15 RFSL	ESAI_RCR Receiver Frame Sync Length. The RFSL bit selects the length of the receive frame sync to be generated or recognized. If RFSL is cleared, a word-length frame sync is selected. If RFSL is set, a 1-bit clock period frame sync is selected. Refer to Figure 52-30 for examples of frame length selection.
14–10 RSWS[4:0]	ESAI_RCR Receiver Slot and Word Select. The RSWS4-RSWS0 bits are used to select the length of the slot and the length of the data words being received through the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 52-35. See also the ESAI data path programming model in Figure 52-2 and Figure 52-3.
9–8 RMOD[1:0]	ESAI_RCR Receiver Network Mode Control. The RMOD1 and RMOD0 bits are used to define the network mode of the ESAI receivers, as shown in Table 52-34. In the normal mode, the frame rate divider determines the word transfer rate - one word is transferred per frame sync during the frame sync time slot, as shown in Figure 52-29. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 52-29. For more details, see Modes of Operation. In order to comply with AC-97 specifications, RSWS4-RSWS0 should be set to 0x00011 (20-bit slot, 20-bit word); RFSL and RFSR should be cleared, and RDC4-RDC0 should be set to 0x0C (13 words in frame).
7 RWA	ESAI_RCR Receiver Word Alignment Control. The Receiver Word Alignment Control (RWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If RWA is cleared, the data word is assumed to be left-aligned in the slot frame. If RWA is set, the data word is assumed to be right-aligned in the slot frame. If the data word is shorter than the slot length, the data bits which are not in the data word field are ignored. For data word lengths of less than 24 bits, the data word is right-extended with zeroes before being stored in the receive data registers.
6 RSHFD	ESAI_RCR Receiver Shift Direction. The RSHFD bit causes the receiver shift registers to shift data in MSB first when RSHFD is cleared or LSB first when RSHFD is set (see Figure 52-2 and Figure 52-3).
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 RE3	ESAI_RCR ESAI Receiver 3 Enable. When RE3 is set and TE2 is cleared, the ESAI receiver 3 is enabled and samples data at the SDO2/SDI3 pin. ESAI_TX2 and ESAI_RX3 should not be enabled at the same time (RE3=1 and TE2=1). When RE3 is cleared, receiver 3 is disabled by inhibiting data transfer into ESAI_RX3. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX3 data register. If RE3 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX3 will be invalid and must be discarded.
2 RE2	ESAI_RCR ESAI Receiver 2 Enable. When RE2 is set and TE3 is cleared, the ESAI receiver 2 is enabled and samples data at the SDO3/SDI2 pin. ESAI_TX3 and ESAI_RX2 should not be enabled at the same time (RE2=1 and TE3=1). When RE2 is cleared, receiver 2 is disabled by inhibiting data transfer into ESAI_RX2. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX2 data register. If RE2 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX2 will be invalid and must be discarded.

Table continues on the next page...

ESAI_RCR field descriptions (continued)

Field	Description
1 RE1	ESAI_RCR ESAI Receiver 1 Enable. When RE1 is set and TE4 is cleared, the ESAI receiver 1 is enabled and samples data at the SDO4/SDI1 pin. ESAI_TX4 and ESAI_RX1 should not be enabled at the same time (RE1=1 and TE4=1). When RE1 is cleared, receiver 1 is disabled by inhibiting data transfer into ESAI_RX1. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX1 data register. If RE1 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX1 will be invalid and must be discarded.
0 RE0	ESAI_RCR ESAI Receiver 0 Enable. When RE0 is set and TE5 is cleared, the ESAI receiver 0 is enabled and samples data at the SDO5/SDI0 pin. ESAI_TX5 and ESAI_RX0 should not be enabled at the same time (RE0=1 and TE5=1). When RE0 is cleared, receiver 0 is disabled by inhibiting data transfer into ESAI_RX0. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the ESAI_RX0 data register. If RE0 is set while some of the other receivers are already in operation, the first data word received in ESAI_RX0 will be invalid and must be discarded.

52.5.17 Receive Clock Control Register (ESAI_RCCR)

The read/write Receiver Clock Control Register (ESAI_RCCR) controls the ESAI receiver clock generator bit and frame sync rates, word length, and number of words per frame for the serial data. The ESAI_RCCR control bits are described in the following paragraphs.

NOTE

ARM Core clock is ipg_clk_esai in block ESAI which is from CCM's ahb_clk_root.

Table 52-37. Receiver High Frequency Clock Divider

RFP3-RFP0	Divide Ratio
0x0	1
0x1	2
0x2	3
0x3	4
...	...
0xF	16

Table 52-38. SCKR Pin Definition Table

Control Bits		SCKR PIN
SYN	RCKD	
0	0	SCKR input

Table continues on the next page...

Table 52-38. SCKR Pin Definition Table (continued)

Control Bits		SCKR PIN
SYN	RCKD	
0	1	SCKR output
1	0	IF0
1	1	OF0

Table 52-39. FSR Pin Definition Table

Control Bits			FSR Pin
SYN	TEBE	RFSD	
0	X	0	FSR input
0	X	1	FSR output
1	0	0	IF1
1	0	1	OF1
1	1	0	reserved
1	1	1	Transmitter Buffer Enable

Table 52-40. HCKR Pin Definition Table

Control Bits		HCKR PIN
SYN	RHCKD	
0	0	HCKR input
0	1	HCKR output
1	0	IF2
1	1	OF2

Address: 4006_2000h base + E0h offset = 4006_20E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								RHCKD	RFSD	RCKD	RHCKP	RFSP	RCKP	RFP[3:0]	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RFP[3:0]		RDC[4:0]					RPSR	RPM[7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_RCCR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 RHCKD	ESAI_RCCR Receiver High Frequency Clock Direction. The Receiver High Frequency Clock Direction (RHCKD) bit selects the source of the receiver high frequency clock when in the asynchronous mode (SYN=0) and the IF2/OF2 flag direction in the synchronous mode (SYN=1). In the asynchronous mode, when RHCKD is set, the internal clock generator becomes the source of the receiver high frequency clock and is the output on the HCKR pin. In the asynchronous mode, when RHCKD is cleared, the receiver high frequency clock source is external; the internal clock generator is disconnected from the HCKR pin, and an external clock source may drive this pin. When RHCKD is cleared, HCKR is an input; when RHCKD is set, HCKR is an output. In the synchronous mode when RHCKD is set, the HCKR pin becomes the OF2 output flag. If RHCKD is cleared, the HCKR pin becomes the IF2 input flag. Refer to Table 52-2 and Table 52-40.
22 RFSD	ESAI_RCCR Receiver Frame Sync Signal Direction. The Receiver Frame Sync Signal Direction (RFSD) bit selects the source of the receiver frame sync signal when in the asynchronous mode (SYN=0) and the IF1/OF1/Transmitter Buffer Enable flag direction in the synchronous mode (SYN=1). In the asynchronous mode, when RFSD is set, the internal clock generator becomes the source of the receiver frame sync and is the output on the FSR pin. In the asynchronous mode, when RFSD is cleared, the receiver frame sync source is external; the internal clock generator is disconnected from the FSR pin, and an external clock source may drive this pin. In the synchronous mode when RFSD is set, the FSR pin becomes the OF1 output flag or the Transmitter Buffer Enable, according to the TEBE control bit. If RFSD is cleared, the FSR pin becomes the IF1 input flag. Refer to Table 52-2 and Table 52-39 .
21 RCKD	ESAI_RCCR Receiver Clock Source Direction. The Receiver Clock Source Direction (RCKD) bit selects the source of the clock signal used to clock the receive shift register in the asynchronous mode (SYN=0) and the IF0/OF0 flag direction in the synchronous mode (SYN=1). In the asynchronous mode, when RCKD is set, the internal clock source becomes the bit clock for the receive shift registers and word length divider and is the output on the SCKR pin. In the asynchronous mode when RCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKR pin, and an external clock source may drive this pin. In the synchronous mode when RCKD is set, the SCKR pin becomes the OF0 output flag. If RCKD is cleared, the SCKR pin becomes the IF0 input flag. Refer to Table 52-2 and Table 52-38.
20 RHCKP	ESAI_RCCR Receiver High Frequency Clock Polarity. The Receiver High Frequency Clock Polarity (RHCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RHCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive high frequency bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RHCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.
19 RFSP	ESAI_RCCR Receiver Frame Sync Polarity. The Receiver Frame Sync Polarity (RFSP) determines the polarity of the receive frame sync signal. When RFSP is cleared the frame sync signal polarity is positive, that is, the frame start is indicated by a high level on the frame sync pin. When RFSP is set the frame sync signal polarity is negative, that is, the frame start is indicated by a low level on the frame sync pin.
18 RCKP	The Receiver Clock Polarity (RCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.
17–14 RFP[3:0]	ESAI_RCCR Rx High Frequency Clock Divider. The RFP3-RFP0 bits control the divide ratio of the receiver high frequency clock to the receiver serial bit clock when the source of the receiver high frequency clock and the bit clock is the internal Arm Core clock. When the HCKR input is being driven

Table continues on the next page...

ESAI_RCCR field descriptions (continued)

Field	Description
	from an external high frequency clock, the RFP3-RFP0 bits specify an additional division ration in the clock divider chain. Table 52-37 provides the specification of the divide ratio. Figure 52-32 shows the ESAI high frequency generator functional diagram.
13–9 RDC[4:0]	ESAI_RCCR Rx Frame Rate Divider Control. The RDC4-RDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the receiver frame clocks. In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (RDC[4:0]=0x00001 to 0x11111) for network mode. A divide ratio of one (RDC[4:0]=0x00000) in network mode is a special case (on-demand mode). In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (RDC[4:0]=0x00000 to 0x11111) for normal mode. In normal mode, a divide ratio of one (RDC[4:0]=0x00000) provides continuous periodic data word transfers. A bit-length frame sync (RFSL=1) must be used in this case. The ESAI frame sync generator functional diagram is shown in Figure 52-33 .
8 RPSR	ESAI_RCCR Receiver Prescaler Range. The RPSR controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When RPSR is set, the fixed prescaler is bypassed. When RPSR is cleared, the fixed divide-by-eight prescaler is operational (see Figure 52-32). The maximum internally generated bit clock frequency is Fsys/4, the minimum internally generated bit clock frequency is Fsys/(2 x 8 x 256 x 16)=Fsys/65536. (Do not use the combination RPSR=1 and RPM7-RPM0 =0x00, which causes synchronization problems when using the internal Core clock as source (RHCKD=1 or RCKD=1))
7–0 RPM[7:0]	ESAI_RCCR Receiver Prescale Modulus Select. The RPM7-RPM0 bits specify the divide ratio of the prescale divider in the ESAI receiver clock generator. A divide ratio from 1 to 256 (RPM[7:0]=0x00 to 0xFF) may be selected. The bit clock output is available at the receiver serial bit clock (SCKR) pin. The bit clock output is also available internally for use as the bit clock to shift the receive shift registers. The ESAI receive clock generator functional diagram is shown in Figure 52-32.

52.5.18 Transmit Slot Mask Register A (ESAI_TSMA)

The Transmit Slot Mask Register A together with Transmit Slot Mask Register B (ESAI_TSMA and ESAI_TSMB) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. Fields ESAI_TSMA [TS[15:0]] and ESAI_TSMB [TS[31:16]] are concatenated to form the 32-bit field TS[31:0]. Bit number n in TS[31:0] is the enable/disable control bit for transmission in slot number n.

Address: 4006_2000h base + E4h offset = 4006_20E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TS[15:0]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESAI_TSMA field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 TS[15:0]	When bit number N in ESAI_TSMA is cleared, all the transmit data pins of the enabled transmitters are tri-stated during transmit time slot number N. The data is still transferred from the transmit data registers to the transmit shift registers but neither the TDE nor the TUE flags are set. This means that during a disabled slot, no transmitter empty interrupt is generated. The Core is interrupted only for enabled slots. Data that is written to the transmit data registers when servicing this request is transmitted in the next enabled transmit time slot. When bit number N in ESAI_TSMA register is set, the transmit sequence is as usual: data is transferred from the TX registers to the shift registers and transmitted during slot number N, and the TDE flag is set. Using the slot mask in ESAI_TSMA does not conflict with using TSR. Even if a slot is enabled in ESAI_TSMA, the user may choose to write to TSR instead of writing to the transmit data registers TXn. This causes all the transmit data pins of the enabled transmitters to be tri-stated during the next slot. Data written to the ESAI_TSMA affects the next frame transmission. The frame being transmitted is not affected by this data and would comply to the last ESAI_TSMA setting. Data read from ESAI_TSMA returns the last written data. After hardware or software reset, the ESAI_TSMA register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data transmission. When operating in normal mode, bit 0 of the ESAI_TSMA register must be set, otherwise no output is generated.

52.5.19 Transmit Slot Mask Register B (ESAI_TSMB)

The Transmit Slot Mask Register B together with Transmit Slot Mask Register A (ESAI_TSMA and ESAI_TSMB) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. Fields ESAI_TSMA [TS[15:0]] and ESAI_TSMB [TS[31:16]] are concatenated to form the 32-bit field TS[31:0]. Bit number n in TS[31:0] is the enable/disable control bit for transmission in slot number n.

Address: 4006_2000h base + E8h offset = 4006_20E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TS[31:16]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESAI_TSMB field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ESAI_TSMB field descriptions (continued)

Field	Description
15–0 TS[31:16]	When bit number N in ESAI_TSMB is cleared, all the transmit data pins of the enabled transmitters are tri-stated during transmit time slot number N. The data is still transferred from the transmit data registers to the transmit shift registers but neither the TDE nor the TUE flags are set. This means that during a disabled slot, no transmitter empty interrupt is generated. The Core is interrupted only for enabled slots. Data that is written to the transmit data registers when servicing this request is transmitted in the next enabled transmit time slot. When bit number N in ESAI_TSMB register is set, the transmit sequence is as usual: data is transferred from the TX registers to the shift registers and transmitted during slot number N, and the TDE flag is set. Using the slot mask in ESAI_TSMB does not conflict with using TSR. Even if a slot is enabled in TSMB, the user may chose to write to TSR instead of writing to the transmit data registers TXn. This causes all the transmit data pins of the enabled transmitters to be tri-stated during the next slot. Data written to the ESAI_TSMB affects the next frame transmission. The frame being transmitted is not affected by this data and would comply to the last ESAI_TSMB setting. Data read from ESAI_TSMB returns the last written data. After hardware or software reset, the ESAI_TSMB register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data transmission.

52.5.20 Receive Slot Mask Register A (ESAI_RSMA)

The Receive Slot Mask Register A together with Receive Slot Mask Register B (ESAI_RSMA and ESAI_RSMB) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. Fields ESAI_RSMA [RS[15:0]] and ESAI_RSMB [RS31:16]] are concatenated to form the 32-bit field RS[31:0]. Bit number n in RS[31:0] is an enable/disable control bit for receiving data in slot number n.

Address: 4006_2000h base + ECh offset = 4006_20ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RS[15:0]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESAI_RSMA field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 RS[15:0]	When bit number N in the ESAI_RSMA register is cleared, the data from the enabled receivers input pins are shifted into their receive shift registers during slot number N. The data is not transferred from the receive shift registers to the receive data registers, and neither the RDF nor the ROE flag is set. This

Table continues on the next page...

ESAI_RSMA field descriptions (continued)

Field	Description
	means that during a disabled slot, no receiver full interrupt is generated. The Core is interrupted only for enabled slots. When bit number N in the ESAI_RSMA is set, the receive sequence is as usual: data which is shifted into the enabled receivers shift registers is transferred to the receive data registers and the RDF flag is set. Data written to the ESAI_RSMA affects the next received frame. The frame being received is not affected by this data and would comply to the last ESAI_RSMA setting. Data read from ESAI_RSMA returns the last written data. After hardware or software reset, the ESAI_RSMA register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data reception. When operating in normal mode, bit 0 of the ESAI_RSMA register must be set to one, otherwise no input is received.

52.5.21 Receive Slot Mask Register B (ESAI_RSMB)

The Receive Slot Mask Register B together with Receive Slot Mask Register A (ESAI_RSMA and ESAI_RSMB) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. Fields ESAI_RSMA [RS[15:0]] and ESAI_RSMB [RS31:16]] are concatenated to form the 32-bit field RS[31:0]. Bit number n in RS[31:0] is an enable/disable control bit for receiving data in slot number n.

Address: 4006_2000h base + F0h offset = 4006_20F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RS[31:16]															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESAI_RSMB field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–0 RS[31:16]	When bit number N in the ESAI_RSMB register is cleared, the data from the enabled receivers input pins are shifted into their receive shift registers during slot number N. The data is not transferred from the receive shift registers to the receive data registers, and neither the RDF nor the ROE flag is set. This means that during a disabled slot, no receiver full interrupt is generated. The Core is interrupted only for enabled slots. When bit number N in the ESAI_RSMB is set, the receive sequence is as usual: data which is shifted into the enabled receivers shift registers is transferred to the receive data registers and the RDF flag is set.

Table continues on the next page...

ESAI_RSMB field descriptions (continued)

Field	Description
	Data written to the ESAI_RSMB affects the next received frame. The frame being received is not affected by this data and would comply to the last ESAI_RSMB setting. Data read from ESAI_RSMB returns the last written data. After hardware or software reset, the ESAI_RSMB register is preset to 0x0000FFFF, which means that all 16 possible slots are enabled for data reception.

52.5.22 Port C Direction Register (ESAI_PRRC)

There are two registers to control the ESAI personal reset status: Port C Direction Register (ESAI_PRRC) and Port C Control Register (ESAI_PCRC).

The read/write 32-bit Port C Direction Register (ESAI_PRRC) in conjunction with the Port C Control Register (ESAI_PCRC) controls the functionality of the ESAI personal reset state. Table 52-47 provides the port pin configurations. Hardware and software reset clear all ESAI_PRRC bits.

Address: 4006_2000h base + F8h offset = 4006_20F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				PDC[11:0]											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_PRRC field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 PDC[11:0]	See Table 52-47.

52.5.23 Port C Control Register (ESAI_PCRC)

The read/write 32-bit Port C Control Register (ESAI_PCRC) in conjunction with the Port C Direction Register (ESAI_PRRC) controls the functionality of the ESAI personal reset state. Each of the PC(11:0) bits controls the functionality of the corresponding port pin. Table 52-47 provides the port pin configurations. Hardware and software reset clear all ESAI_PCRC bits.

Table 52-47. PCRC and PRRC Bits Functionality

PDC[i]	PC[i]	Port Pin[i] Function
0	0	Disconnected
1	1	ESAI

Address: 4006_2000h base + FCh offset = 4006_20FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				PC[11:0]											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESAI_PCRC field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 PC[11:0]	See Table 52-47.

Chapter 53
Asynchronous Sample Rate Converter (ASRC)

53.1 Introduction

The figure below is a system view of the connection between the ASRC block and other blocks.

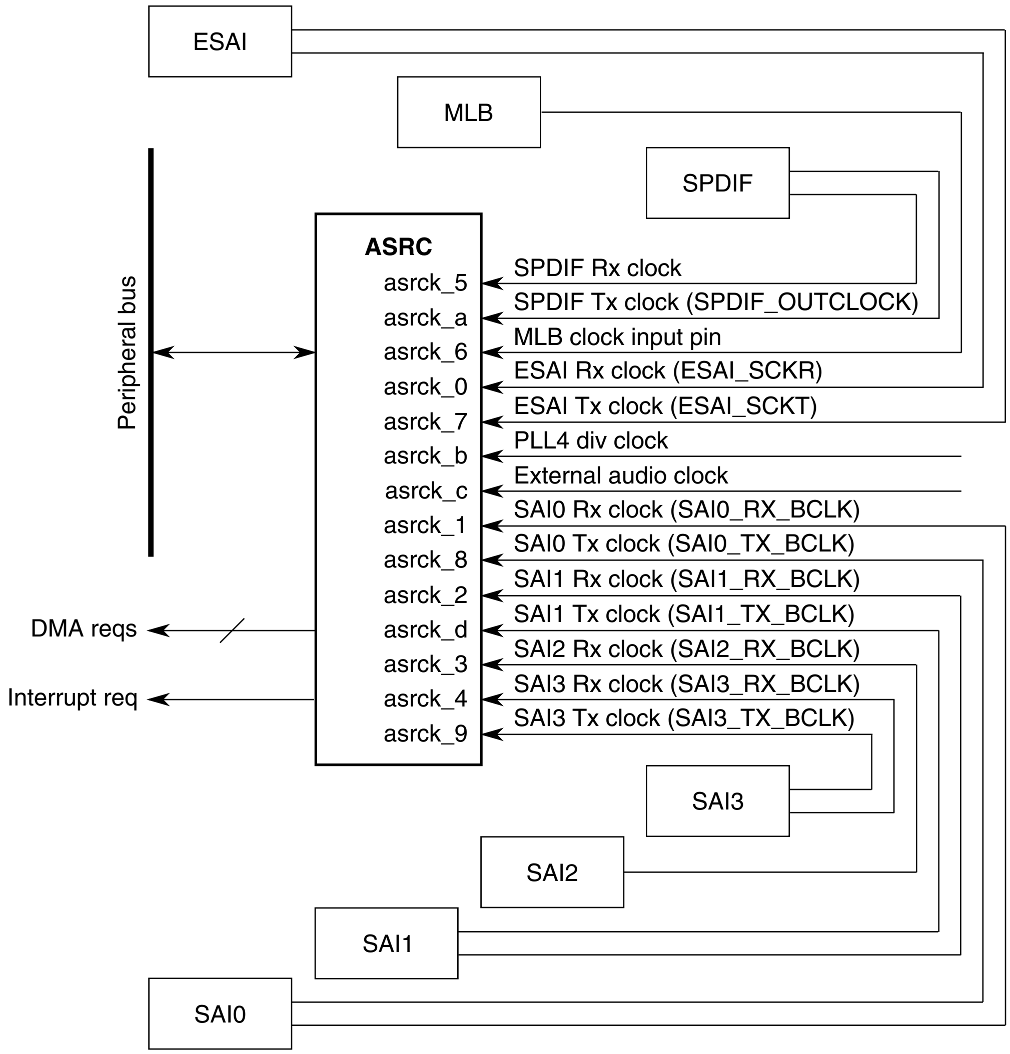


Figure 53-1. System Overview

The following figure is the ASRC block diagram.

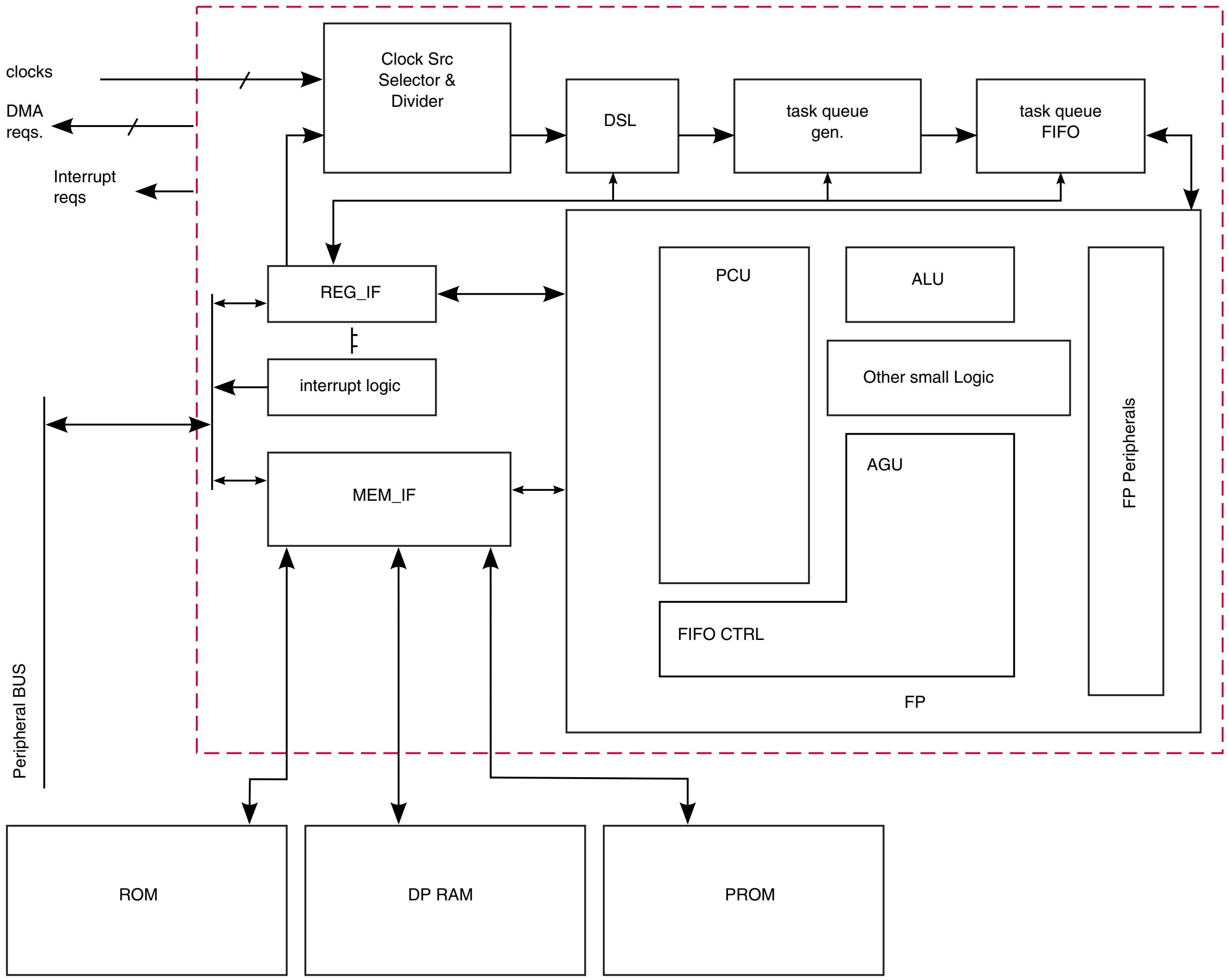


Figure 53-2. ASRC block diagram

53.1.1 Overview

The Asynchronous Sample Rate Converter (ASRC) converts the sampling rate of a signal associated with an input clock into a signal associated with a different output clock. The ASRC supports concurrent sample rate conversion of up to 10 channels of about -120dB THD+N. The ASRC supports up to three sampling rate pairs.

The incoming audio data to this chip may be received from various sources at different sampling rates. The outgoing audio data of this chip may have different sampling rates and it can also be associated with output clocks that are asynchronous to the input clocks.

The ASRC is implemented as a co-processor in hardware, with minimal ARM Platform intervention required.

53.1.2 Features

Features:

- Any number (0-10) of contiguous channels can be associated to one of the sampling rate pairs.
- Support user-programmable threshold for the input/output FIFOs.
- Support flexible 8/16/24 bit width of input data, and 16/24 bit width of output data.
- Designed for rate conversion between 44.1kHz, 32kHz, 48kHz, 96kHz, and 192kHz. The useful signal bandwidth is below 24kHz.
- Other input sampling rates in the range of 8kHz to 200kHz is also supported, but possibly with less desirable bandwidth.
- Other output sampling rates in the range of 30kHz to 200kHz is also supported, but possibly with less desirable bandwidth.
- Limited support for the case when output sampling rates is between 8kHz and 30kHz. The limitation is the supported ratio (Fsin/Fsout) range as between 1/24 to 8
- Automatic accommodation to slow variations in the incoming and outgoing sampling rates.
- Linear phase
- Tolerant to sample clock jitter
- Designed for real-time streaming audio usage. The output sampling clock must be always physically available in the system.

Clock/Data Connections

- The sampling rate clocks are directly connected to the ASRC block, the ratio estimation of the input clocks with output clocks are done in ASRC hardware.
- The clock signals come from the following blocks, for example:
  - ESAI, receiving bit clock and transmitting bit clock
  - SAI, receiving bit clock and transmitting bit clock
  - SPDIF, receiving bit clock and transmitting bit clock
  - other audio peripherals etc.
- The exchange of audio data is done by the processor accessing ASRC block through registers defined on shared peripheral bus.

53.1.3 Modes of Operation

See the Programmable Registers section for a definition of the registers and parameters used in ASRC.

53.1.3.1 Data Transfer Schemes

53.1.3.1.1 Data Input Modes

The input mode for each of the three channel sets may be set independently. Three modes of supplying data to the ASRC input FIFOs are available:

- Polling
- Interrupt
- DMA

In all input-data transfer schemes, the ASRC fetches data from each enabled FIFO and processes the data sample-by-sample after each rising edge of the associated input sampling clock until the FIFO level reaches a threshold.

After the threshold is reached, the ASRC requests data. The FIFO size for each channel set is 64 samples and the threshold is set at 32 samples. The threshold can be defined by interface registers ASRMCRx, x=A, B or C.

If the ASRC attempts to fetch data from an empty FIFO, an error is generated and the ASRSTR_AOLE bit is set. If the ASRC overload interrupt is enabled (ASRIER_AOLIE bit is set), an interrupt is generated.

When writing data to an input FIFO, you must ensure that it is in a predefined sequence. For example, when writing to an input FIFO, the sequence should be: channel_0, channel_1, channel_2,..., channel_n, channel_0, channel_1, channel_2, etc. Here channel_n stands for the data intended for the n-th channel. The hardware will re-allocate each data to its corresponding channel FIFO. The channel being re-allocated is shown by ASRCCR_ACIA.

Mode 1 (Polling Mode)

Polling mode is the default mode following power-on or individual reset, and is selected by clearing the associated channel set A, B, or C data-input interrupt enable bit (ASRIER_ADIEx, where x=A, B or C). In this mode, data-input interrupts are disabled. When the FIFO level is below the threshold, the associated status bit (ASRSTR_AIDEx, where x=A, B, or C) is set. To clear the status bit, the FIFO must be written with enough data to raise the level above the threshold.

Mode 2 (Interrupt Mode)

The ASRC input FIFOs can also be serviced by interrupts. To enable interrupts, the corresponding data-input interrupt enable bits (ASRIER_ADIEx, where x=A, B, or C) should be set. An interrupt is automatically generated any time the input FIFO level is below the threshold. The interrupt is cleared when enough data is written to the FIFO to raise the level above the threshold.

Mode 3 (DMA Mode)

The ASRC input FIFOs can also be filled using DMA. In this mode, the data-input interrupt-enable bits (ASRIER_ADIEx, where x=A, B, or C) should be cleared and the DMA controller should be configured to use the ASRC as a request source.

53.1.3.1.2 Data Output Modes

The output mode for each of the 3 channel sets (A, B, and C) may be set independently.

Three modes of retrieving data from the ASRC output FIFOs are available:

- Polling
- Interrupt
- DMA

In all output-data transfer schemes, the ASRC places a processed sample into the associated output FIFO. After a threshold is reached, the ASRC requests that data be transferred out of the FIFO.

The FIFO size for each channel set is 64 samples and the threshold is set at 32 samples. The threshold can be defined by interface registers ASRMCRx, x=A, B or C.

If the ASRC attempts to place data into a FIFO that is already full, an error is generated and the ASRSTR_AOLE bit is set. If the ASRC overload interrupt is enabled (ASRIER_AOLIE bit is set), an interrupt is generated.

Each output FIFO is organized in the same channel order in which the associated input FIFO was written.

Two transfer modes are supported by Interface Block.

Mode 1 (Polling Mode)

The ASRC output FIFOs can be serviced by polling. In this mode, ensure the associated output-data interrupt enable bit (ASRIER_ADOEx, where x=A, B, or C) is cleared. In this mode, all output-data interrupts are disabled. Any time the output FIFO exceeds the threshold the associated status bit (ASRSTR_AODFx, where x=A, B, or C) is set. To clear the status bit, enough data must be read from the associated output FIFO to lower the level below the threshold.

Mode 2 (Interrupt Mode)

The ASRC output FIFOs may also be serviced using interrupts. To enable this mode, the corresponding output-data interrupt-enable bits (ASRIER_ADOEx, where x=A, B, or C) should be set. Any time the output FIFO level exceeds the threshold, an interrupt is automatically generated. The interrupt is cleared when enough data is read from the FIFO to lower the level below the threshold.

Mode 3 (DMA Mode)

The ASRC output FIFOs can also be read using DMA. In this mode, the output-data interrupt-enable bits (ASRIER_ADOEx, where x=A, B, or C) should be cleared and the DMA controller should be configured to use the ASRC as a request source.

53.1.3.2 Word Alignment Supported

53.1.3.2.1 Input Data Alignment Modes

The position and length of input data word to the input data FIFOs ASRDIA, ASRDIB, ASRDIC are programmable. The control bits are defined in ASRMCR1x {x=A, B, or C}. It supports the following modes.

Table 53-1. Input Data Alignment

Format	Bit Number																														
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
8-bit LSB Aligned																									7	6	5	4	3	2	1	0
8-bit MSB Aligned																	7	6	5	4	3	2	1	0								
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

53.1.3.2.2 Output Data Alignment Modes

The position and length of output data word from the output data FIFOs ASRDOA, ASRDOB, ASRDOC are programmable. The control bits are defined in ASRMCR1x {x=A, B, or C}. It supports the following modes.

Table 53-2. Output Data Alignment

Format	Bit Number																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit LSB Aligned																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit LSB Aligned with Sign Extension	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16-bit MSB Aligned	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
24-bit LSB Aligned									23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit LSB Aligned with Sign Extension	23	23	23	23	23	23	23	23	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
24-bit MSB Aligned	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								

53.2 Interrupts

ASRC has several interrupts events.

Priority	Description
lowest	ASRC Pair A input data needed
	ASRC Pair B input data needed
	ASRC Pair C input data needed
	ASRC Pair A output data ready
	ASRC Pair B output data ready
	ASRC Pair C output data ready
	ASRC Overload

53.3 DMA requests

ASRC has six DMA requests. They are directly connected to the lowest six status bits in the ASRSTR register.

Table 53-4. DMA requests

Type	Description
0	ASRC Pair A input data needed
1	ASRC Pair B input data needed
2	ASRC Pair C input data needed
3	ASRC Pair A output data ready
4	ASRC Pair B output data ready
5	ASRC Pair C output data ready

53.4 Programmable Registers

All useful registers are listed in the memory map below. The access of undefined registers will behave as normal registers.

All the interface registers are LSB aligned except the input FIFOs and the output FIFOs, and each register has only 24 effective bits.

The input FIFO and output FIFO word alignment can be defined using ASRMCR1{A,B,C} registers in 32-bit interface system.

ASRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_0000	ASRC Control Register (ASRC_ASRCTR)	32	R/W	0000_0000h	53.4.1/2956
4006_0004	ASRC Interrupt Enable Register (ASRC_ASRIER)	32	R/W	0000_0000h	53.4.2/2958
4006_000C	ASRC Channel Number Configuration Register (ASRC_ASRCNCR)	32	R/W	0000_0000h	53.4.3/2960
4006_0010	ASRC Filter Configuration Status Register (ASRC_ASRCFG)	32	R/W	0000_0000h	53.4.4/2962
4006_0014	ASRC Clock Source Register (ASRC_ASRCSR)	32	R/W	0000_0000h	53.4.5/2964
4006_0018	ASRC Clock Divider Register 1 (ASRC_ASRCDR1)	32	R/W	0000_0000h	53.4.6/2968
4006_001C	ASRC Clock Divider Register 2 (ASRC_ASRCDR2)	32	R/W	0000_0000h	53.4.7/2969

Table continues on the next page...

ASRC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_0020	ASRC Status Register (ASRC_ASRSTR)	32	R	0000_0000h	53.4.8/2970
4006_0040	ASRC Parameter Register n (ASRC_ASRPMn1)	32	R/W	0000_0000h	53.4.9/2973
4006_0044	ASRC Parameter Register n (ASRC_ASRPMn2)	32	R/W	0000_0000h	53.4.9/2973
4006_0048	ASRC Parameter Register n (ASRC_ASRPMn3)	32	R/W	0000_0000h	53.4.9/2973
4006_004C	ASRC Parameter Register n (ASRC_ASRPMn4)	32	R/W	0000_0000h	53.4.9/2973
4006_0050	ASRC Parameter Register n (ASRC_ASRPMn5)	32	R/W	0000_0000h	53.4.9/2973
4006_0054	ASRC ASRC Task Queue FIFO Register 1 (ASRC_ASRTFR1)	32	R/W	0000_0000h	53.4.10/ 2974
4006_005C	ASRC Channel Counter Register (ASRC_ASRCCR)	32	R/W	0000_0000h	53.4.11/ 2975
4006_0060	ASRC Data Input Register for Pair x (ASRC_ASRDIA)	32	W	0000_0000h	53.4.12/ 2976
4006_0064	ASRC Data Output Register for Pair x (ASRC_ASRDOA)	32	R	0000_0000h	53.4.13/ 2976
4006_0068	ASRC Data Input Register for Pair x (ASRC_ASRDIB)	32	W	0000_0000h	53.4.12/ 2976
4006_006C	ASRC Data Output Register for Pair x (ASRC_ASRDOB)	32	R	0000_0000h	53.4.13/ 2976
4006_0070	ASRC Data Input Register for Pair x (ASRC_ASRDIC)	32	W	0000_0000h	53.4.12/ 2976
4006_0074	ASRC Data Output Register for Pair x (ASRC_ASRDOC)	32	R	0000_0000h	53.4.13/ 2976
4006_0080	ASRC Ideal Ratio for Pair A-High Part (ASRC_ASRIDRHA)	32	R/W	0000_0000h	53.4.14/ 2977
4006_0084	ASRC Ideal Ratio for Pair A -Low Part (ASRC_ASRIDRLA)	32	R/W	0000_0000h	53.4.15/ 2977
4006_0088	ASRC Ideal Ratio for Pair B-High Part (ASRC_ASRIDRHB)	32	R/W	0000_0000h	53.4.16/ 2978
4006_008C	ASRC Ideal Ratio for Pair B-Low Part (ASRC_ASRIDRLB)	32	R/W	0000_0000h	53.4.17/ 2978
4006_0090	ASRC Ideal Ratio for Pair C-High Part (ASRC_ASRIDRHC)	32	R/W	0000_0000h	53.4.18/ 2979
4006_0094	ASRC Ideal Ratio for Pair C-Low Part (ASRC_ASRIDRLC)	32	R/W	0000_0000h	53.4.19/ 2979
4006_0098	ASRC 76kHz Period in terms of ASRC processing clock (ASRC_ASR76K)	32	R/W	0000_0A47h	53.4.20/ 2980
4006_009C	ASRC 56kHz Period in terms of ASRC processing clock (ASRC_ASR56K)	32	R/W	0000_0DF3h	53.4.21/ 2981
4006_00A0	ASRC Misc Control Register for Pair A (ASRC_ASRMCRA)	32	R/W	0000_0000h	53.4.22/ 2982
4006_00A4	ASRC FIFO Status Register for Pair A (ASRC_ASRFSTA)	32	R	0000_0000h	53.4.23/ 2984

Table continues on the next page...

ASRC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_00A8	ASRC Misc Control Register for Pair B (ASRC_ASRMCRB)	32	R/W	0000_0000h	53.4.24/ 2985
4006_00AC	ASRC FIFO Status Register for Pair B (ASRC_ASRFSTB)	32	R	0000_0000h	53.4.25/ 2987
4006_00B0	ASRC Misc Control Register for Pair C (ASRC_ASRMCRC)	32	R/W	0000_0000h	53.4.26/ 2988
4006_00B4	ASRC FIFO Status Register for Pair C (ASRC_ASRFSTC)	32	R	0000_0000h	53.4.27/ 2990
4006_00C0	ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1A)	32	R/W	0000_0000h	53.4.28/ 2991
4006_00C4	ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1B)	32	R/W	0000_0000h	53.4.28/ 2991
4006_00C8	ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1C)	32	R/W	0000_0000h	53.4.28/ 2991

53.4.1 ASRC Control Register (ASRC_ASRCTR)

The ASRC control register (ASRCTR) is a 24-bit read/write register that controls the ASRC operations.

Address: 4006_0000h base + 0h offset = 4006_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								Reserved	ATSC	ATSB	ATSA	Reserved	USRC	IDRC	USRB
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IDRB	USRA	IDRA	Reserved									ASREC	ASREB	ASREA	ASRCEN
W												SRST				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCTR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 -	This field is reserved. Reserved. Should be written as zero for compatibility.
22 ATSC	ASRC Pair C Automatic Selection For Processing Options When this bit is 1, pair C will automatic update its pre-processing and post-processing options (ASRCFG: PREMODC, ASRCFG:POSTMODC) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly. When this bit is 0, the user needs to choose the proper processing options for pair C. This bit should be disabled when {USRC, IDRC}={1,1}.
21 ATSB	ASRC Pair B Automatic Selection For Processing Options When this bit is 1, pair B will automatic update its pre-processing and post-processing options (ASRCFG: PREMODB, ASRCFG:POSTMODB) based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly. When this bit is 0, the user is responsible for choosing the proper processing options for pair B. This bit should be disabled when {USRB, IDRB}={1,1}.
20 ATSA	ASRC Pair A Automatic Selection For Processing Options When this bit is 1, pair A will automatic update its pre-processing and post-processing options (ASRCFG: PREMODA, ASRCFG:POSTMODA)based on the frequencies it detected. To use this option, the two parameter registers(ASR76K and ASR56K) should be set correctly. When this bit is 0, the user needs to choose the proper processing options for pair A. This bit should be disabled when {USRA, IDRA}={1,1}.
19 -	This field is reserved. Reserved. Should be written as zero for compatibility.
18 USRC	Use Ratio for Pair C Use ratio as the input to ASRC. This bit is used in conjunction with IDRC control bit.
17 IDRC	Use Ideal Ratio for Pair C When USRC=0, this bit has no usage. When USRC=1 and IDRC=0, ASRC internal measured ratio will be used. When USRC=1 and IDRC=1, the idea ratio from the interface register ASRIDRHC, ASRIDRLC will be used. It is suggested to manually set ASRCFG:POSTMODC, ASRCFG:PREMODC.
16 USRB	Use Ratio for Pair B Use ratio as the input to ASRC. This bit is used in conjunction with IDRB control bit.
15 IDRB	Use Ideal Ratio for Pair B When USRB=0, this bit has no usage. When USRB=1 and IDRB=0, ASRC internal measured ratio will be used. When USRB=1 and IDRB=1, the idea ratio from the interface register ASRIDRHB, ASRIDRLB will be used.It is suggested to manually set ASRCFG:POSTMODB, ASRCFG:PREMODB.
14 USRA	Use Ratio for Pair A Use ratio as the input to ASRC. This bit is used in conjunction with IDRA control bit.

Table continues on the next page...

ASRC_ASRCTR field descriptions (continued)

Field	Description
13 IDRA	Use Ideal Ratio for Pair A When USRA=0, this bit has no usage. When USRA=1 and IDRA=0, ASRC internal measured ratio will be used. When USRA=1 and IDRA=1, the idea ratio from the interface register ASRIDRHA, ASRIDRLA will be used. It is suggested to manually set ASRCFG:POSTMODA, ASRCFG:PREMODA.
12–5 -	This field is reserved. Reserved. Should be written as zero for compatibility.
4 SRST	Software Reset This bit is self-clear bit. Once it is been written as 1, it will generate a software reset signal inside ASRC. After 9 cycles of the ASRC processing clock, this reset process will stop, and this bit will be cleared automatically.
3 ASREC	ASRC Enable C Enable the operation of the conversion C of ASRC. When ASREC is cleared, operation of conversion C is disabled.
2 ASREB	ASRC Enable B Enable the operation of the conversion B of ASRC. When ASREB is cleared, operation of conversion B is disabled.
1 ASREA	ASRC Enable A Enable the operation of the conversion A of ASRC. When ASREA is cleared, operation of conversion A is disabled.
0 ASRCEN	ASRC Enable Enable the operation of ASRC.

53.4.2 ASRC Interrupt Enable Register (ASRC_ASRIER)

Address: 4006_0000h base + 4h offset = 4006_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved								AFPWE	AOLIE	ADOEC	ADOEB	ADOEA	ADIEC	ADIEB	ADIEA
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIER field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–8 -	This field is reserved. Reserved. Should be written as zero for compatibility.
7 AFPWE	FP in Wait State Interrupt Enable Enables the FP in wait state interrupt. 1 interrupt enabled 0 interrupt disabled
6 AOLIE	Overload Interrupt Enable Enables the overload interrupt. 1 interrupt enabled 0 interrupt disabled
5 ADOEC	Data Output C Interrupt Enable Enables the data output C interrupt. 1 interrupt enabled 0 interrupt disabled
4 ADOEB	Data Output B Interrupt Enable Enables the data output B interrupt. 1 interrupt enabled 0 interrupt disabled
3 ADOEA	Data Output A Interrupt Enable Enables the data output A interrupt. 1 interrupt enabled 0 interrupt disabled
2 ADIEC	Data Input C Interrupt Enable Enables the data input C interrupt. 1 interrupt enabled 0 interrupt disabled
1 ADIEB	Data Input B Interrupt Enable Enables the data input B interrupt. 1 interrupt enabled 0 interrupt disabled
0 ADIEA	Data Input A Interrupt Enable Enables the data input A Interrupt. 1 interrupt enabled 0 interrupt disabled

53.4.3 ASRC Channel Number Configuration Register (ASRC_ASRCNCR)

The ASRC channel number configuration register (ASRCNCR) is a 24-bit read/write register that sets the number of channels used by each ASRC conversion pair.

There are 10 channels available for distribution among 3 conversion pairs, they are ordered as 0,1,...,9. The bottom [0, ANCA-1] channels are used for pair A, the top [10-ANCC, 9] channels are used for pair C, and the [ANCA, ANCA+ANCB-1] channels are allocated for pair B. In case that ANCA=0, then the [0, ANCB-1] channels are assigned for pair B.

Address: 4006_0000h base + Ch offset = 4006_000Ch

Bit	31–24	23–12	11–8	7–4	3–0
R/W	Reserved	Reserved	ANCC	ANCB	ANCA
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

ASRC_ASRCNCR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–12 -	This field is reserved. Reserved. Should be written as zero for compatibility.
11–8 ANCC	Number of C Channels [1] 0000 0 channels in C (Pair C is disabled) 0001 1 channel in C 0010 2 channels in C 0011 3 channels in C 0100 4 channels in C 0101 5 channels in C 0110 6 channels in C 0111 7 channels in C 1000 8 channels in C 1001 9 channels in C 1010 10 channels in C 1011-1111 Should not be used.
7–4 ANCB	Number of B Channels 0000 0 channels in B (Pair B is disabled) 0001 1 channel in B 0010 2 channels in B 0011 3 channels in B 0100 4 channels in B 0101 5 channels in B

Table continues on the next page...

ASRC_ASRCNCR field descriptions (continued)

Field	Description
	0110 6 channels in B 0111 7 channels in B 1000 8 channels in B 1001 9 channels in B 1010 10 channels in B 1011-1111 Should not be used.
3–0 ANCA	Number of A Channels 0000 0 channels in A (Pair A is disabled) 0001 1 channel in A 0010 2 channels in A 0011 3 channels in A 0100 4 channels in A 0101 5 channels in A 0110 6 channels in A 0111 7 channels in A 1000 8 channels in A 1001 9 channels in A 1010 10 channels in A 1011-1111 Should not be used.

1. ANCC+ANCB+ANCA<=10. Hardware is not checking the constraint. Programmer should take the responsibility to ensure the constraint is satisfied.

53.4.4 ASRC Filter Configuration Status Register (ASRC_ASRCFG)

The ASRC configuration status register (ASRCFG) is a 24-bit read/write register that sets and/or automatically senses the ASRC operations.

Address: 4006_0000h base + 10h offset = 4006_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								INIRQC	INIRQB	INIRQA	NDPRC	NDPRB	NDPRA	POSTMODC[1-0]	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PREMODC[1-0]		POSTMODB[1-0]		PREMODB[1-0]		POSTMODA[1-0]		PREMODA[1-0]		Reserved					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCFG field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.

Table continues on the next page...

ASRC_ASRCFG field descriptions (continued)

Field	Description
23 INIRQC	Initialization for Conversion Pair C is served When this bit is 1, it means the initialization for conversion pair C is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREC=0 or ASRCTR:ASRCEN=0).
22 INIRQB	Initialization for Conversion Pair B is served When this bit is 1, it means the initialization for conversion pair B is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREB=0 or ASRCTR:ASRCEN=0).
21 INIRQA	Initialization for Conversion Pair A is served When this bit is 1, it means the initialization for conversion pair A is served. This bit is cleared by disabling the ASRC conversion pair (ASRCTR:ASREA=0 or ASRCTR:ASRCEN=0).
20 NDPRC	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair C 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Don't use default parameters for RAM-stored parameters. Use the parameters already stored in RAM.
19 NDPRB	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair B 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Don't use default parameters for RAM-stored parameter. Use the parameters already stored in RAM.
18 NDPRA	Not Use Default Parameters for RAM-stored Parameters For Conversion Pair A 0 Use default parameters for RAM-stored parameters. Override any parameters already in RAM. 1 Don't use default parameters for RAM-stored parameters. Use the parameters already stored in RAM.
17–16 POSTMODC[1-0]	Post-Processing Configuration for Conversion Pair C These bits will be read/write by user if ASRCTR:ATSC=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSC=1. These bits set the selection of the post-processing configuration. 00 Select Upsampling-by-2 as defined in Signal Processing Flow. 01 Select Direct-Connection as defined in Signal Processing Flow. 10 Select Downsampling-by-2 as defined in Signal Processing Flow.
15–14 PREMODC[1-0]	Pre-Processing Configuration for Conversion Pair C These bits will be read/write by user if ASRCTR:ATSC=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSC=1. These bits set the selection of the pre-processing configuration. 00 Select Upsampling-by-2 01 Select Direct-Connection 10 Select Downsampling-by-2 11 Select passthrough mode. In this case, POSTMODC[1-0] have no use.
13–12 POSTMODB[1-0]	Post-Processing Configuration for Conversion Pair B These bits will be read/write by user if ASRCTR:ATSB=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSB=1. These bits set the selection of the post-processing configuration. 00 Select Upsampling-by-2

Table continues on the next page...

ASRC_ASRCFG field descriptions (continued)

Field	Description
	01 Select Direct-Connection 10 Select Downsampling-by-2
11–10 PREMODB[1-0]	Pre-Processing Configuration for Conversion Pair B These bits will be read/write by user if ASRCTR:ATSB=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSB=1. These bits set the selection of the pre-processing configuration. 00 Select Upsampling-by-2 01 Select Direct-Connection 10 Select Downsampling-by-2 11 Select passthrough mode. In this case, POSTMODB[1-0] have no use.
9–8 POSTMODA[1-0]	Post-Processing Configuration for Conversion Pair A These bits will be read/write by user if ASRCTR:ATSA=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSA=1. These bits set the selection of the post-processing configuration. 00 Select Upsampling-by-2 01 Select Direct-Connection 10 Select Downsampling-by-2
7–6 PREMODA[1-0]	Pre-Processing Configuration for Conversion Pair A These bits will be read/write by user if ASRCTR:ATSA=0, and can also be automatically updated by the ASRC internal logic if ASRCTR:ATSA=1. These bits set the selection of the pre-processing configuration. 00 Select Upsampling-by-2 01 Select Direct-Connection 10 Select Downsampling-by-2 11 Select passthrough mode. In this case, POSTMODA[1-0] have no use.
5–0 -	This field is reserved. Reserved. Should be written as zero for compatibility.

53.4.5 ASRC Clock Source Register (ASRC_ASRCSR)

The ASRC clock source register (ASRCSR) is a 24-bit read/write register that controls the sources of the input and output clocks of the ASRC.

Table 53-11. Bit Clock Definitions

Bit Clk Name	Definitions
0	ESAI RX clock
1	SAI0 RX clock
2	SAI1 RX clock
3	SAI2 RX clock
4	SAI3 RX clock

Table continues on the next page...

Table 53-11. Bit Clock Definitions (continued)

Bit Clk Name	Definitions
5	SPDIF RX clock
6	Reserved
7	ESAI Tx Clock
8	SAI0 TX clock
9	SAI3 TX clock
a	SPDIF Tx Clock
b	Divided PLL4 Main clock
c	Audio External Clock-in
d	SAI1 Tx Clock

Address: 4006_0000h base + 14h offset = 4006_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								AOCSC				AOCSB				AOCSA				AICSC				AICSB				AICSA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCSR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–20 AOCSC	**Output Clock Source C** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0
19–16 AOCSB	**Output Clock Source B** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2

Table continues on the next page...

ASRC_ASRCSR field descriptions (continued)

Field	Description
	0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0
15–12 AOCSA	**Output Clock Source A** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0
11–8 AICSC	**Input Clock Source C** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D

Table continues on the next page...

ASRC_ASRCSR field descriptions (continued)

Field	Description
	1111 clock disabled, connected to zero any other value bit clock 0
7–4 AICSB	**Input Clock Source B** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0
3–0 AICSA	**Input Clock Source A** 0000 bit clock 0 0001 bit clock 1 0010 bit clock 2 0011 bit clock 3 0100 bit clock 4 0101 bit clock 5 0110 bit clock 6 0111 bit clock 7 1000 bit clock 8 1001 bit clock 9 1010 bit clock A 1011 bit clock B 1100 bit clock C 1101 bit clock D 1111 clock disabled, connected to zero any other value bit clock 0

53.4.6 ASRC Clock Divider Register 1 (ASRC_ASRCDR1)

The ASRC clock divider register (ASRCDR1) is a two 24-bit read/write register that controls the division factors of the ASRC input and output clock sources.

Address: 4006_0000h base + 18h offset = 4006_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								AOCDB			AOCPB			AOCDA			AOCPA			AICDB			AICPB			AICDA			AICPA		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCDR1 field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–21 AOCDB	Output Clock Divider B Specify the divide ratio of the output clock divider B. The divide ratio may range from 1 to 8 (AOCDB[2:0] = 000 to 111).
20–18 AOCPB	Output Clock Prescaler B Specify the prescaling factor of the output prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
17–15 AOCDA	Output Clock Divider A Specify the divide ratio of the output clock divider A. The divide ratio may range from 1 to 8 (AOCDA[2:0] = 000 to 111).
14–12 AOCPA	Output Clock Prescaler A Specify the prescaling factor of the output prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.
11–9 AICDB	Input Clock Divider B Specify the divide ratio of the input clock divider B. The divide ratio may range from 1 to 8 (AICDB[2:0] = 000 to 111).
8–6 AICPB	Input Clock Prescaler B Specify the prescaling factor of the input prescaler B. The prescaling ratio may be any power of 2 from 1 to 128.
5–3 AICDA	Input Clock Divider A Specify the divide ratio of the input clock divider A. The divide ratio may range from 1 to 8 (AICDA[2:0] = 000 to 111).
2–0 AICPA	Input Clock Prescaler A Specify the prescaling factor of the input prescaler A. The prescaling ratio may be any power of 2 from 1 to 128.

53.4.7 ASRC Clock Divider Register 2 (ASRC_ASRCDR2)

The ASRC clock divider register (ASRCDR2) is a two 24-bit read/write register that controls the division factors of the ASRC input and output clock sources.

Address: 4006_0000h base + 1Ch offset = 4006_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved												AOCDC			AOCPC			AICDC			AICPC		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCDR2 field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–12 -	This field is reserved. Reserved. Should be written as zero for compatibility.
11–9 AOCDC	Output Clock Divider C Specify the divide ratio of the output clock divider C. The divide ratio may range from 1 to 8 (AOCDC[2:0] = 000 to 111).
8–6 AOCPC	Output Clock Prescaler C Specify the prescaling factor of the output prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.
5–3 AICDC	Input Clock Divider C Specify the divide ratio of the input clock divider C. The divide ratio may range from 1 to 8 (AICDC[2:0] = 000 to 111).
2–0 AICPC	Input Clock Prescaler C Specify the prescaling factor of the input prescaler C. The prescaling ratio may be any power of 2 from 1 to 128.

53.4.8 ASRC Status Register (ASRC_ASRSTR)

The ASRC status register (ASRSTR) is a 24-bit read-write register used by the processor core to examine the status of the ASRC block and clear the overload interrupt request and AOLE flag bit. Read the status register will return the current state of ASRC.

Address: 4006_0000h base + 20h offset = 4006_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								Reserved		DSLCNT	ATQOL	AOOLC	AOOLB	AOOLA	AIOLC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	AIOLB	AIOLA	AODOC	AODOB	AODOA	AIDUC	AIDUB	AIDUA	FPWT	AOLE	AODFC	AODFB	AODFA	AIDEC	AIDEB	AIDEA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRSTR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.

Table continues on the next page...

ASRC_ASRSTR field descriptions (continued)

Field	Description
23–22 -	This field is reserved. Reserved. Should be written as zero for compatibility.
21 DSLCNT	DSL Counter Input to FIFO ready When set, this bit indicates that new DSL counter information is stored in the internal ASRC FIFO. When clear, this bit indicates that new DSL counter information is in the process of storage into the internal ASRC FIFO. When ASRIER:AFPWE=1, the rising edge of this signal will propose an interrupt request. Writing any value with this bit set will clear the interrupt request proposed by the rising edge of this bit.
20 ATQOL	Task Queue FIFO overload When set, this bit indicates that task queue FIFO logic is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
19 AOOLC	Pair C Output Task Overload When set, this bit indicates that pair C output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
18 AOOLB	Pair B Output Task Overload When set, this bit indicates that pair B output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
17 AOOLA	Pair A Output Task Overload When set, this bit indicates that pair A output task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
16 AIOLC	Pair C Input Task Overload When set, this bit indicates that pair C input task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
15 AIOLB	Pair B Input Task Overload When set, this bit indicates that pair B input task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
14 AIOLA	Pair A Input Task Overload When set, this bit indicates that pair A input task is oveloaded. This may help to check the reason why overload interrupt happens. The bit is cleared when writing ASRCTR:AOLIE as 1.
13 AODOC	Output Data Buffer C has overflowed When set, this bit indicates that output data buffer C has overflowed. When clear, this bit indicates that output data buffer C has not overflowed
12 AODOB	Output Data Buffer B has overflowed When set, this bit indicates that output data buffer B has overflowed. When clear, this bit indicates that output data buffer B has not overflowed

Table continues on the next page...

ASRC_ASRSTR field descriptions (continued)

Field	Description
11 AODOA	Output Data Buffer A has overflowed When set, this bit indicates that output data buffer A has overflowed. When clear, this bit indicates that output data buffer A has not overflowed
10 AIDUC	Input Data Buffer C has underflowed When set, this bit indicates that input data buffer C has underflowed. When clear, this bit indicates that input data buffer C has not underflowed.
9 AIDUB	Input Data Buffer B has underflowed When set, this bit indicates that input data buffer B has underflowed. When clear, this bit indicates that input data buffer B has not underflowed.
8 AIDUA	Input Data Buffer A has underflowed When set, this bit indicates that input data buffer A has underflowed. When clear, this bit indicates that input data buffer A has not underflowed.
7 FPWT	FP is in wait states This bit is for debug only. When set, this bit indicates that ASRC is in wait states. When clear, this bit indicates that ASRC is not in wait states. If ASRCTR:AFPWE=1 and ASRCTR:ASDBG=1, an interrupt will be proposed when this bit is set.
6 AOLE	Overload Error Flag When set, this bit indicates that the task rate is too high for the ASRC to handle. The reasons for overload may be: - too high input clock frequency, - too high output clock frequency, - incorrect selection of the pre-filter, - low ASRC processing clock, - too many channels, - underrun, - or any combination of the reasons above. Since the ASRC uses the same hardware resources to perform various tasks, the real reason for the overload is not straight forward, and it should be carefully analyzed by the programmer. If ASRCTR:AOLIE=1, an interrupt will be proposed when this bit is set. Write any value with this bit set as one into the status register will clear this bit and the interrupt request proposed by this bit.
5 AODFC	Number of data in Output Data Buffer C is greater than threshold When set, this bit indicates that number of data already existing in ASRDORC is greater than threshold and the processor can read data from ASRDORC. When AODFC is set, the ASRC generates data output C interrupt request to the processor, if enabled (that is, ASRCTR:ADOEC = 1). A DMA request is always generated when the AODFC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
4 AODFB	Number of data in Output Data Buffer B is greater than threshold

Table continues on the next page...

ASRC_ASRSTR field descriptions (continued)

Field	Description
	When set, this bit indicates that number of data already existing in ASRDORB is greater than threshold and the processor can read data from ASRDORB. When AODFB is set, the ASRC generates data output B interrupt request to the processor, if enabled (that is, ASRCTR:ADOEB = 1). A DMA request is always generated when the AODFB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
3 AODFA	Number of data in Output Data Buffer A is greater than threshold When set, this bit indicates that number of data already existing in ASRDORA is greater than threshold and the processor can read data from ASRDORA. When AODFA is set, the ASRC generates data output A interrupt request to the processor, if enabled (that is, ASRCTR:ADOEA = 1). A DMA request is always generated when the AODFA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
2 AIDEC	Number of data in Input Data Buffer C is less than threshold When set, this bit indicates that number of data still available in ASRDIRC is less than threshold and the processor can write data to ASRDIRC. When AIDEC is set, the ASRC generates data input C interrupt request to the processor, if enabled (that is, ASRCTR:ADIEC = 1). A DMA request is always generated when the AIDEC bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
1 AIDEB	Number of data in Input Data Buffer B is less than threshold When set, this bit indicates that number of data still available in ASRDIRB is less than threshold and the processor can write data to ASRDIRB. When AIDEB is set, the ASRC generates data input B interrupt request to the processor, if enabled (that is, ASRCTR:ADIEB = 1). A DMA request is always generated when the AIDEB bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.
0 AIDEA	Number of data in Input Data Buffer A is less than threshold When set, this bit indicates that number of data still available in ASRDIRA is less than threshold and the processor can write data to ASRDIRA. When AIDEA is set, the ASRC generates data input A interrupt request to the processor, if enabled (that is, ASRCTR:ADIEA = 1). A DMA request is always generated when the AIDEA bit is set, but a DMA transfer takes place only if a DMA channel is active and triggered by this event.

53.4.9 ASRC Parameter Register n (ASRC_ASRPMn*n*)

Parameter registers determine the performance of ASRC.

The parameter registers must be initialized by software before ASRC is enabled.

Table 53-16. ASRC Parameter Registers (ASRPM1~ASRPM5)

Register	Offset	Access	Reset Value	Recommend Value
asrcpm1	0x40	R/W	0x00_0000	0x7fffff
asrcpm2	0x44	R/W	0x00_0000	0x255555
asrcpm3	0x48	R/W	0x00_0000	0xff7280
asrcpm4	0x4C	R/W	0x00_0000	0xff7280
asrcpm5	0x50	R/W	0x00_0000	0xff7280

Address: 4006_0000h base + 40h offset + (4d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								PARAMETER_VALUE																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRPMn*n* field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 PARAMETER_ VALUE	See recommended values table.

53.4.10 ASRC ASRC Task Queue FIFO Register 1 (ASRC_ASRTFR1)

The register defines and shows the parameters for ASRC inner task queue FIFOs.

Address: 4006_0000h base + 54h offset = 4006_0054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Reserved				TF_FILL							TF_BASE							Reserved					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRTFR1 field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–20 -	This field is reserved. Reserved. Should be written as zero for compatibility.
19–13 TF_FILL	Current number of entries in task queue FIFO.
12–6 TF_BASE	Base address for task queue FIFO. Set to 0x7C.
5–0 -	This field is reserved. Reserved. Should be written as zero for compatibility.

53.4.11 ASRC Channel Counter Register (ASRC_ASRCCR)

The ASRC channel counter register (ASRCCR) is a 24-bit read/write register that sets and reflects the current specific input/output FIFO being accessed through shared peripheral bus for each ASRC conversion pair.

Address: 4006_0000h base + 5Ch offset = 4006_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								ACOC				ACOB				ACOA				ACIC				ACIB				ACIA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRCCR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–20 ACOC	The channel counter for Pair C's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair C's output FIFO's usage. The value can be any value between [0, ANCC-1]
19–16 ACOB	The channel counter for Pair B's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair B's output FIFO's usage. The value can be any value between [0, ANCB-1]
15–12 ACOA	The channel counter for Pair A's output FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair A's output FIFO's usage. The value can be any value between [0, ANCA-1]
11–8 ACIC	The channel counter for Pair C's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair C's input FIFO's usage. The value can be any value between [0, ANCC-1]
7–4 ACIB	The channel counter for Pair B's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair B's input FIFO's usage. The value can be any value between [0, ANCB-1]
3–0 ACIA	The channel counter for Pair A's input FIFO These bits stand for the current channel being accessed through shared peripheral bus for Pair A's input FIFO's usage. The value can be any value between [0, ANCA-1]

53.4.12 ASRC Data Input Register for Pair *x* (ASRC_ASRDI*n*)

These registers are the interface registers for the audio data input of pair A,B,C respectively. They are backed by FIFOs.

Address: 4006_0000h base + 60h offset + (8d × i), where i=0d to 2d

Bit	31–24	23–0
R	Reserved	
W		DATA
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

ASRC_ASRDI*n* field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 DATA	Audio data input

53.4.13 ASRC Data Output Register for Pair *x* (ASRC_ASRDO*n*)

These registers are the interface registers for the audio data output of pair A,B,C respectively. They are backed by FIFOs.

Address: 4006_0000h base + 64h offset + (8d × i), where i=0d to 2d

Bit	31–24	23–0
R	Reserved	DATA
W		
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

ASRC_ASRDO*n* field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 DATA	Audio data output

53.4.14 ASRC Ideal Ratio for Pair A-High Part (ASRC_ASRIDRHA)

The ideal ratio registers (ASRIDRHA, ASRIDRLA) hold the ratio value IDRATIOA. IDRATIOA = Fs $_{inA}$ /Fs $_{outA}$ = Ts $_{outA}$ /Ts $_{inA}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRA, IDRA}=2'b11.

Address: 4006_0000h base + 80h offset = 4006_0080h

Bit	31–24	23–16	15–8	7–0
R/W	Reserved	Reserved	Reserved	IDRATIOA[31:24]
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

ASRC_ASRIDRHA field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–8 -	This field is reserved. Reserved
7–0 IDRATIOA[31:24]	IDRATIOA[31:24]. High part of ideal ratio value for pair A

53.4.15 ASRC Ideal Ratio for Pair A -Low Part (ASRC_ASRIDRLA)

The ideal ratio registers (ASRIDRHA, ASRIDRLA) hold the ratio value IDRATIOA. IDRATIOA = Fs $_{inA}$ /Fs $_{outA}$ = Ts $_{outA}$ /Ts $_{inA}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRA, IDRA}=2'b11.

Address: 4006_0000h base + 84h offset = 4006_0084h

Bit	31–24	23–0
R/W	Reserved	IDRATIOA[23:0]
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

ASRC_ASRIDRLA field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 IDRATIOA[23:0]	IDRATIOA[23:0]. Low part of ideal ratio value for pair A

53.4.16 ASRC Ideal Ratio for Pair B-High Part (ASRC_ASRIDRHB)

The ideal ratio registers (ASRIDRHB, ASRIDRLB) hold the ratio value IDRATIOB. IDRATIOB = $Fs_{inB} / Fs_{outB} = Ts_{outB} / Ts_{inB}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRB, IDRB}=2'b11.

Address: 4006_0000h base + 88h offset = 4006_0088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved																IDRATIOB[31:24]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIDRHB field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–8 -	This field is reserved. Reserved
7–0 IDRATIOB[31:24]	IDRATIOB[31:24]. High part of ideal ratio value for pair B.

53.4.17 ASRC Ideal Ratio for Pair B-Low Part (ASRC_ASRIDRLB)

The ideal ratio registers (ASRIDRHB, ASRIDRLB) hold the ratio value IDRATIOB. IDRATIOB = $Fs_{inB} / Fs_{outB} = Ts_{outB} / Ts_{inB}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRB, IDRB}=2'b11.

Address: 4006_0000h base + 8Ch offset = 4006_008Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								IDRATIOB[23:0]																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIDRLB field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 IDRATIOB[23:0]	IDRATIOB[23:0]. Low part of ideal ratio value for pair B.

53.4.18 ASRC Ideal Ratio for Pair C-High Part (ASRC_ASRIDRHC)

The ideal ratio registers (ASRIDRHC, ASRIDRLC) hold the ratio value IDRATIOC. IDRATIOC = $Fs_{inC} / Fs_{outC} = Ts_{outC} / Ts_{inC}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRC, IDRC}=2'b11.

Address: 4006_0000h base + 90h offset = 4006_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved																IDRATIOC[31:24]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIDRHC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–8 -	This field is reserved. Reserved
7–0 IDRATIOC[31:24]	IDRATIOC[31:24]. High part of ideal ratio value for pair C.

53.4.19 ASRC Ideal Ratio for Pair C-Low Part (ASRC_ASRIDRLC)

The ideal ratio registers (ASRIDRHC, ASRIDRLC) hold the ratio value IDRATIOC. IDRATIOC = $Fs_{inC} / Fs_{outC} = Ts_{outC} / Ts_{inC}$ is a 32-bit fixed point value with 26 fractional bits. This value is only useful when ASRCTR:{USRC, IDRC}=2'b11.

Address: 4006_0000h base + 94h offset = 4006_0094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								IDRATIOC[23:0]																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRIDRLC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 IDRATIOC[23:0]	IDRATIOC[23:0]. Low part of ideal ratio value for pair C.

53.4.20 ASRC 76kHz Period in terms of ASRC processing clock (ASRC_ASR76K)

The register (ASR76K) holds the period of the 76kHz sampling clock in terms of the ASRC processing clock with frequency Fs_{ASRC} . ASR76K = Fs_{ASRC} /Fs_{76k} . Reset value is 0x0A47 which assumes that Fs_{ASRC} =200MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically. In a system when Fs_{ASRC} =133MHz, the value should be assigned explicitly as 0x06D6 in user application code.

Address: 4006_0000h base + 98h offset = 4006_0098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved							ASR76K																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	1

ASRC_ASR76K field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–17 -	This field is reserved. Reserved
16–0 ASR76K	Value for the period of the 76kHz sampling clock.

53.4.21 ASRC 56kHz Period in terms of ASRC processing clock (ASRC_ASR56K)

The register (ASR56K) holds the period of the 56kHz sampling clock in terms of the ASRC processing clock with frequency Fs_{ASRC} . ASR56K = Fs_{ASRC} /Fs_{56k} . Reset value is 0x0DF3 which assumes that Fs_{ASRC} =200MHz. This register is used to help the ASRC internal logic to decide the pre-processing and the post-processing options automatically. In a system when Fs_{ASRC} =133MHz, the value should be assigned explicitly as 0x0947 in user application code.

Address: 4006_0000h base + 9Ch offset = 4006_009Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								Reserved							ASR56K																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	1	0	0	1	1

ASRC_ASR56K field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–17 -	This field is reserved. Reserved
16–0 ASR56K	Value for the period of the 56kHz sampling clock

53.4.22 ASRC Misc Control Register for Pair A (ASRC_ASRMCRA)

The register (ASRMCRA) is used to control Pair A internal logic.

Address: 4006_0000h base + A0h offset = 4006_00A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								ZEROBUFA	EXTTHRSHA	BUFSTALLA	BYPASSPOLYA	Reserved		OUTFIFO_THRESHOLDA[5:0]	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	OUTFIFO_THRESHOLDA[5:0]				RSYNIFA	RSYNOFA	Reserved				INFIFO_THRESHOLDA[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCRA field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 ZEROBUFA	Initialize buf of Pair A when pair A is enabled. Always clear option. This bit is used to control whether the buffer is to be zeroized when pair A is enabled. 1 Don't zeroize the buffer 0 Zeroize the buffer
22 EXTTHRSHA	Use external thresholds for FIFO control of Pair A This bit will determine whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair A. 1 Use external defined thresholds. 0 Use default thresholds.
21 BUFSTALLA	Stall Pair A conversion in case of Buffer Near Empty/Full Condition This bit will determine whether the near empty/full FIFO condition will stall the rate conversion for pair A. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel.

Table continues on the next page...

ASRC_ASRMCRA field descriptions (continued)

Field	Description
	1 Stall Pair A conversion in case of near empty/full FIFO conditions. 0 Don't stall Pair A conversion even in case of near empty/full FIFO conditions.
20 BYPASSPOLYA	Bypass Polyphase Filtering for Pair A This bit will determine whether the polyphase filtering part of Pair A conversion will be bypassed. 1 Bypass polyphase filtering. 0 Don't bypass polyphase filtering.
19–18 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
17–12 OUTFIFO_ THRESHOLDA[5:0]	The threshold for Pair A's output FIFO per channel These bits stand for the threshold for Pair A's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFA	Re-sync Input FIFO Channel Counter If bit set, force ASRCCR:ACIA=0. If bit clear, untouch ASRCCR:ACIA.
10 RSYNOFA	Re-sync Output FIFO Channel Counter If bit set, force ASRCCR:ACOA=0. If bit clear, untouch ASRCCR:ACOA.
9–6 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
5–0 INFIFO_ THRESHOLDA[5:0]	The threshold for Pair A's input FIFO per channel These bits stand for the threshold for Pair A's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared. **NOTE:** This field is writable only if EXTHRSHA is set.

53.4.23 ASRC FIFO Status Register for Pair A (ASRC_ASRFSTA)

The register (ASRFSTA) is used to show Pair A internal FIFO conditions.

Address: 4006_0000h base + A4h offset = 4006_00A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								OAFA	Reserved				OUTFIFO_ FILLA[5:0]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUTFIFO_FILLA[5:0]				IAEA	Reserved				INFIFO_FILLA[5:0]						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRFSTA field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 OAFA	Output FIFO is near Full for Pair A This bit is to indicate whether the output FIFO of Pair A is near full.
22–19 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
18–12 OUTFIFO_ FILLA[5:0]	The fillings for Pair A's output FIFO per channel These bits stand for the fillings for Pair A's output FIFO per channel. Possible range is [0,64].

Table continues on the next page...

ASRC_ASRFSTA field descriptions (continued)

Field	Description
11 IAEA	Input FIFO is near Empty for Pair A This bit is to indicate whether the input FIFO of Pair A is near empty.
10–7 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
6–0 INFIFO_FILLA[5:0]	The fillings for Pair A's input FIFO per channel These bits stand for the fillings for Pair A's input FIFO per channel. Possible range is [0,64].

53.4.24 ASRC Misc Control Register for Pair B (ASRC_ASRMCRB)

The register (ASRMCRB) is used to control Pair B internal logic.

Address: 4006_0000h base + A8h offset = 4006_00A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								ZEROBUFB	EXTTHRSHB	BUFSTALLB	BYPASSPOLYB	Reserved		OUTFIFO_THRESHOLDB[5:0]	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	OUTFIFO_THRESHOLDB[5:0]				RSYNIFB	RSYNOFB	Reserved				INFIFO_THRESHOLDB[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCRB field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 ZEROBUFB	Initialize buf of Pair B when pair B is enabled This bit is used to control whether the buffer is to be zeroized when pair B is enabled. 1 Don't zeroize the buffer 0 Zeroize the buffer
22 EXTTHRSHB	Use external thresholds for FIFO control of Pair B

Table continues on the next page...

ASRC_ASRMCRB field descriptions (continued)

Field	Description
	This bit will determine whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair B. 1 Use external defined thresholds. 0 Use default thresholds.
21 BUFSTALLB	Stall Pair B conversion in case of Buffer Near Empty/Full Condition This bit will determine whether the near empty/full FIFO condition will stall the rate conversion for pair B. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 1 Stall Pair B conversion in case of near empty/full FIFO conditions. 0 Don't stall Pair B conversion even in case of near empty/full FIFO conditions.
20 BYPASSPOLYB	Bypass Polyphase Filtering for Pair B This bit will determine whether the polyphase filtering part of Pair B conversion will be bypassed. 1 Bypass polyphase filtering. 0 Don't bypass polyphase filtering.
19–18 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
17–12 OUTFIFO_ THRESHOLDB[5:0]	The threshold for Pair B's output FIFO per channel These bits stand for the threshold for Pair B's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFB	Re-sync Input FIFO Channel Counter If bit set, force ASRCCR:ACIB=0. If bit clear, untouch ASRCCR:ACIB.
10 RSYNOFB	Re-sync Output FIFO Channel Counter If bit set, force ASRCCR:ACOB=0. If bit clear, untouch ASRCCR:ACOB.
9–6 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
5–0 INFIFO_ THRESHOLDB[5:0]	The threshold for Pair B's input FIFO per channel These bits stand for the threshold for Pair B's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared. **NOTE:** This field is writable only if EXTHRSHB is set.

53.4.25 ASRC FIFO Status Register for Pair B (ASRC_ASRFSTB)

The register (ASRFSTB) is used to show Pair B internal FIFO conditions.

Address: 4006_0000h base + ACh offset = 4006_00ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								OAFB	Reserved				OUTFIFO_FILLB[5:0]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUTFIFO_FILLB[5:0]				IAEB	Reserved				INFIFO_FILLB[5:0]						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRFSTB field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 OAFB	Output FIFO is near Full for Pair B This bit is to indicate whether the output FIFO of Pair B is near full.
22–19 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
18–12 OUTFIFO_ FILLB[5:0]	The fillings for Pair B's output FIFO per channel These bits stand for the fillings for Pair B's output FIFO per channel. Possible range is [0,64].

Table continues on the next page...

ASRC_ASRFSTB field descriptions (continued)

Field	Description
11 IAEB	Input FIFO is near Empty for Pair B This bit is to indicate whether the input FIFO of Pair B is near empty.
10–7 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
6–0 INFIFO_FILLB[5:0]	The fillings for Pair B's input FIFO per channel These bits stand for the fillings for Pair B's input FIFO per channel. Possible range is [0,64].

53.4.26 ASRC Misc Control Register for Pair C (ASRC_ASRMCRC)

The register (ASRMCRC) is used to control Pair C internal logic.

Address: 4006_0000h base + B0h offset = 4006_00B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								ZEROBUFC	EXTTHRSHC	BUFSTALLC	BYPASSPOLYC	Reserved		OUTFIFO_THRESHOLDC[5:0]	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	OUTFIFO_THRESHOLDC[5:0]				RSYNIFC	RSYNOFC	Reserved				INFIFO_THRESHOLDC[5:0]					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCRC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 ZEROBUFC	Initialize buf of Pair C when pair C is enabled This bit is used to control whether the buffer is to be zeroized when pair C is enabled. 1 Don't zeroize the buffer 0 Zeroize the buffer
22 EXTTHRSHC	Use external thresholds for FIFO control of Pair C

Table continues on the next page...

ASRC_ASRMCRC field descriptions (continued)

Field	Description
	This bit will determine whether the FIFO thresholds externally defined in this register is used to control ASRC internal FIFO logic for pair C. 1 Use external defined thresholds. 0 Use default thresholds.
21 BUFSTALLC	Stall Pair C conversion in case of Buffer Near Empty/Full Condition This bit will determine whether the near empty/full FIFO condition will stall the rate conversion for pair C. This option can only work when external ratio is used. Near empty condition is the condition when input FIFO has less than 4 useful samples per channel. Near full condition is the condition when the output FIFO has less than 4 vacant sample words to fill per channel. 1 Stall Pair C conversion in case of near empty/full FIFO conditions. 0 Don't stall Pair C conversion even in case of near empty/full FIFO conditions.
20 BYPASSPOLYC	Bypass Polyphase Filtering for Pair C This bit will determine whether the polyphase filtering part of Pair C conversion will be bypassed. 1 Bypass polyphase filtering. 0 Don't bypass polyphase filtering.
19–18 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
17–12 OUTFIFO_ THRESHOLDC[5:0]	The threshold for Pair C's output FIFO per channel These bits stand for the threshold for Pair C's output FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of output FIFO fillings of the pair is greater than n samples per channel, the output data ready flag is set; when the number of output FIFO fillings of the pair is less than or equal to n samples per channel, the output data ready flag is automatically cleared.
11 RSYNIFC	Re-sync Input FIFO Channel Counter If bit set, force ASRCCR:ACIC=0. If bit clear, untouch ASRCCR:ACIC.
10 RSYNOFC	Re-sync Output FIFO Channel Counter If bit set, force ASRCCR:ACOC=0. If bit clear, untouch ASRCCR:ACOC.
9–6 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
5–0 INFIFO_ THRESHOLDC[5:0]	The threshold for Pair C's input FIFO per channel These bits stand for the threshold for Pair C's input FIFO per channel. Possible range is [0,63]. When the value is n, it means that: when the number of input FIFO fillings of the pair is less than n samples per channel, the input data needed flag is set; when the number of input FIFO fillings of the pair is greater than or equal to n samples per channel, the input data needed flag is automatically cleared. **NOTE:** This field is writable only if EXTHRSHC is set.

53.4.27 ASRC FIFO Status Register for Pair C (ASRC_ASRFSTC)

The register (ASRFSTC) is used to show Pair C internal FIFO conditions.

Address: 4006_0000h base + B4h offset = 4006_00B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								OAFC	Reserved				OUTFIFO_FILLC[5:0]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUTFIFO_FILLC[5:0]				IAEC	Reserved				INFIFO_FILLC[5:0]						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRFSTC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 OAFC	Output FIFO is near Full for Pair C This bit is to indicate whether the output FIFO of Pair C is near full.
22–19 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
18–12 OUTFIFO_ FILLC[5:0]	The fillings for Pair C's output FIFO per channel These bits stand for the fillings for Pair C's output FIFO per channel. Possible range is [0,64].

Table continues on the next page...

ASRC_ASRFSTC field descriptions (continued)

Field	Description
11 IAEC	Input FIFO is near Empty for Pair C This bit is to indicate whether the input FIFO of Pair C is near empty.
10–7 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
6–0 INFIFO_FILLC[5:0]	The fillings for Pair C's input FIFO per channel These bits stand for the fillings for Pair C's input FIFO per channel. Possible range is [0,64].

53.4.28 ASRC Misc Control Register 1 for Pair X (ASRC_ASRMCR1*n*)

The register (ASRMCR1A) is used to control Pair *x* internal logic (for data alignment etc.).

The bit assignment for all the input data formats is the same as that supported by the SSI.

Address: 4006_0000h base + C0h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved				IWD[2:0]			IMSB	Reserved					OMSB	OSGN	OW16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ASRC_ASRMCR1*n* field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–12 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
11–9 IWD[2:0]	Data Width of the input FIFO These three bits will determine the bitwidth for the audio data into ASRC All other settings not shown are reserved. 3'b000 24-bit audio data. 3'b001 16-bit audio data.

Table continues on the next page...

ASRC_ASRMCR1*n* field descriptions (continued)

Field	Description
	3'b010 8-bit audio data.
8 IMSB	Data Alignment of the input FIFO This bit will determine the data alignment of the input FIFO. 1 MSB aligned. 0 LSB aligned.
7–3 -	This field is reserved. Reserved. Should be written as zero for future compatibility.
2 OMSB	Data Alignment of the output FIFO This bit will determine the data alignment of the output FIFO. 1 MSB aligned. 0 LSB aligned.
1 OSGN	Sign Extension Option of the output FIFO This bit will determine the sign extension option of the output FIFO. 1 Sign extension. 0 No sign extension.
0 OW16	Bit Width Option of the output FIFO This bit will determine the bit width option of the output FIFO. 1 16-bit output data 0 24-bit output data.

53.5 Functional Description

53.5.1 Algorithm Description

53.5.1.1 Signal Processing Flow

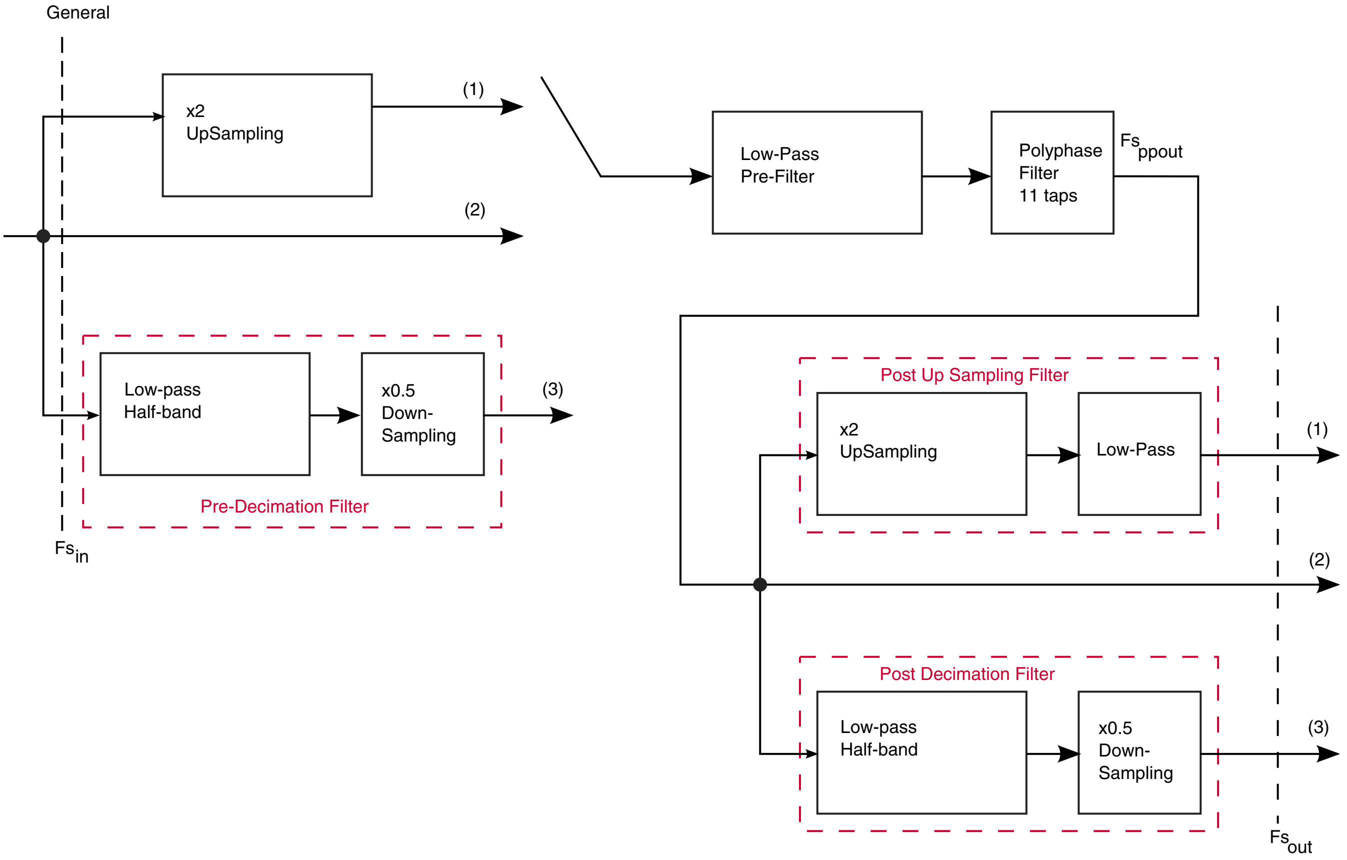


Figure 53-45. Signal processing configurations

The figure above shows the possible configurations of the ASRC. Each configuration consists of 2 to 4 stages.

- x2 up-sampling rate expander (zero insertion only) (Input Branch 1), or direct connection (Input Branch 2), or low-pass pre decimation filter (consisting of a low-pass half-band FIR filter with x0.5 downsampling rate decimator) (Input Branch 3),
- low-pass pre-filter, the low-pass bandwidth is at most 0.25 x Fs, where Fs is the sampling rate of the input signal to this low-pass pre-filter,
- polyphase filter,
- x2 post upsampling filter (consisting of a x2 up-sampling rate expander (zero insertion only) with low-pass half-band FIR filter) (Output Branch 1), or direct connection (Output Branch 2), or low-pass post decimation filter (consisting of a low-pass half-band FIR filter with x0.5 downsampling rate decimator) (Output Branch 3).

By flowing through different processing branches, and different setup of the pre-filter, this ASRC scheme can be used to handle different requirement of rate conversion.

Configuration (a): Input Branch 1+Output Branch 1: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$.

Configuration (b): Input Branch 1+Output Branch 2: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$.

Configuration (c): Input Branch 1+Output Branch 3: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/2$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$.

Configuration (d): Input Branch 2+Output Branch 1: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$.

Configuration (e): Input Branch 2+Output Branch 2: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$.

Configuration (f): Input Branch 2+Output Branch 3: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/4$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$.

Configuration (g): Input Branch 3+Output Branch 1: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}/2$.

Configuration (h): Input Branch 3+Output Branch 2: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = Fs_{out}$.

Configuration (i): Input Branch 3+Output Branch 3: The signal bandwidth observed before the polyphase filter is at most $BW_{in} = Fs_{in}/8$. The signal sampling rate of the polyphase filter output is $Fs_{ppout} = 2Fs_{out}$.

Table 53-55. Pre-Processing, Post-Processing Options

{Pre_Proc, Post_Proc}	Fsout (KHz)								
	8	32	44.1	48	64	88.2	96	128	192

Table continues on the next page...

Table 53-55. Pre-Processing, Post-Processing Options (continued)

Fsin (KHz)	8	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	12	{0,2}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	16	{1,2}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}	{0,0}
	24	{1,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}	{0,0}
	32	{1,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}	{0,0}
	44.1	{2,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}	{0,0}
	48	{2,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}
	64	{2,2}	{0,2}	{0,2}	{0,2}	{0,1}	{0,1}	{0,1}	{0,1}	{0,0}
	88.2	NA	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	96	NA	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	128	NA	{1,2}	{1,2}	{1,2}	{1,1}	{1,1}	{1,1}	{1,1}	{1,1}
	192	NA	{2,2}	{2,2}	{2,2}	{2,1}	{2,1}	{2,1}	{2,1}	{2,1}

Comments:

In the {Pre_Proc, Post_Proc} pair, the meaning of the values are:

Pre_Proc:

- 0 --- Pre-processing Branch 1
- 1 --- Pre-processing Branch 2
- 2 --- Pre-processing Branch 3 decimation-by-2

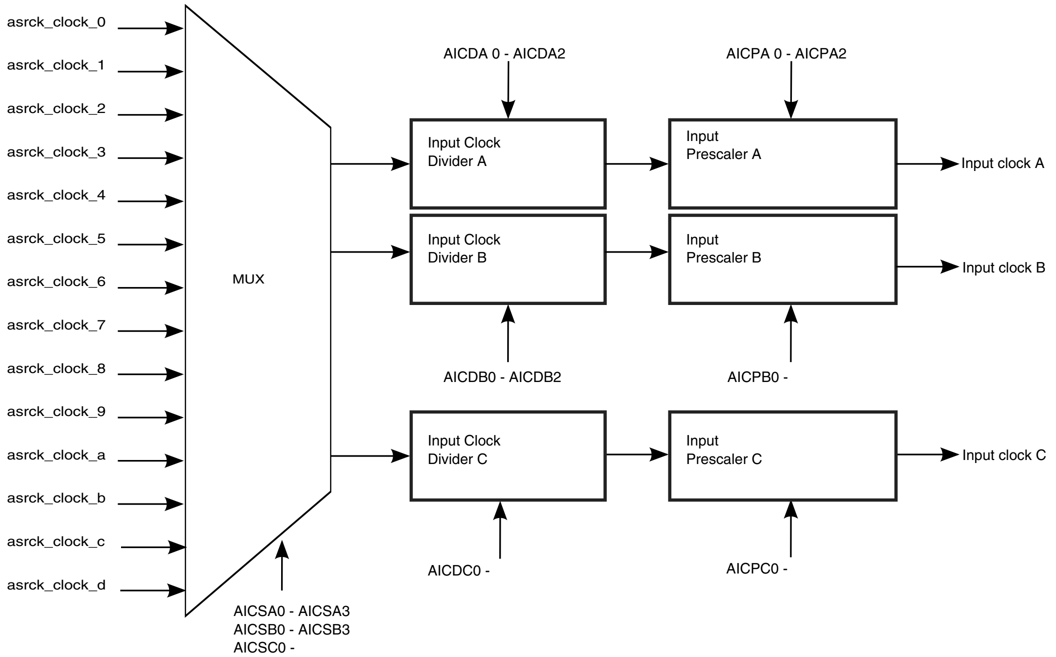
Post_Proc:

- 0 --- Post-processing Branch 1
- 1 --- Post-processing Branch 2
- 2 --- Post-processing Branch 3

53.5.1.2 Operation of the Filter

53.5.1.2.1 Support of Physical Clocks

This design supports physical sampling clocks. The clocks can be provided by Sony/ Phillips digital interface (SPDIF), Enhanced Serial Audio Interface (ESAI), Synchronus Audio Interface. , Core master clock derivative as ASRCK1 .



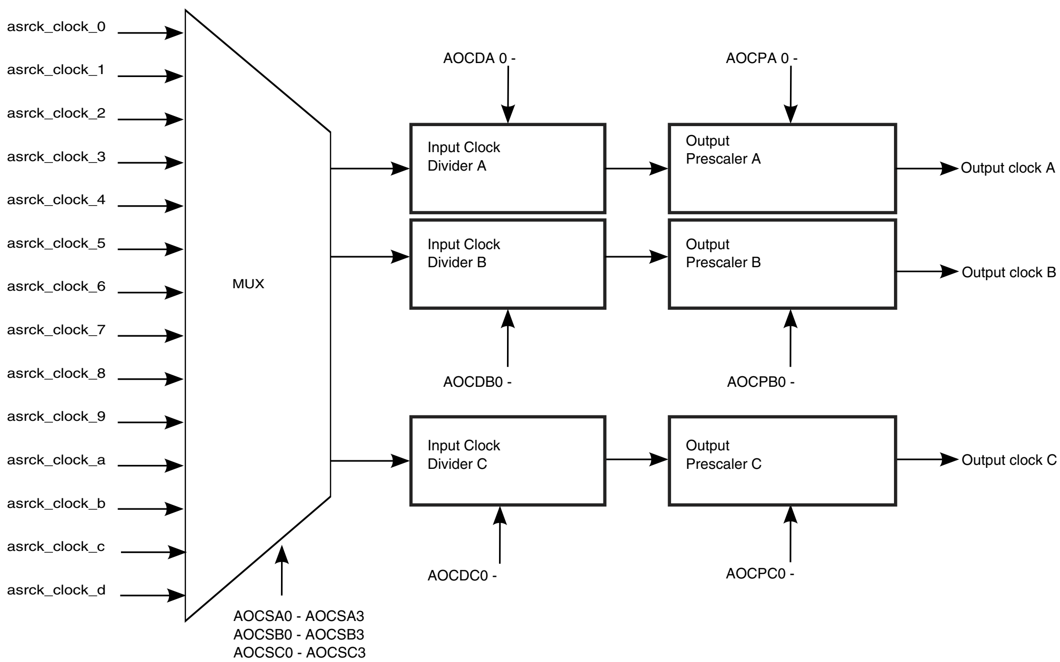


Figure 53-46. Clock Source Selector & Divider

Software can set the ASRC Clock Source Register (ASRCSR) and the Clock Divider Register to select the desired clock source and divide it to the needed sample rate clock for use by the ASRC. The clocks have the following restriction. If the prescaler is set to 1, the clock divider can only be set to 1 and the clock source must have a 50% duty cycle.

53.6 Startup Procedure

The following example shows the normal setup procedure for the ASRC block.

```
#include "asrc_common.h"

#include "stdio.h"

#include "soc_api.h"

int incnt=0;

int outcnt=0;

#include "wy_ideal_ratio_dataini_part.h"

WORD   IdealRatio_High=0x04; //

WORD   IdealRatio_Low=0x0; //

void asrc_config_alloc(WORD ASRCTR_VAL, WORD ASRIER_VAL, WORD ASRCNCR_VAL,
                           WORD ASRCFG_VAL, WORD ASRCDR1_VAL, WORD ASRCDR2_VAL,
                           WORD ASRCSR_VAL)

{  // Disable ASRC

    reg32_write(ASRC_ASRCTR,0x0);

    reg32_write(ASRC_ASRCTR,ASRCTR_VAL);

    reg32_write(ASRC_ASRIER,ASRIER_VAL);

    reg32_write(ASRC_ASRIEM,0x0);

    reg32_write(ASRC_ASRCNCR,ASRCNCR_VAL);

    reg32_write(ASRC_ASRCFG,ASRCFG_VAL);

    reg32_write(ASRC_ASRCDR1,ASRCDR1_VAL);

    reg32_write(ASRC_ASRCDR2,ASRCDR2_VAL);

    reg32_write(ASRC_ASRCSR, ASRCSR_VAL);

    reg32_write(ASRC_ASRPM1, 0x7fffff);

    reg32_write(ASRC_ASRPM2, 0x255555);

    reg32_write(ASRC_ASRPM3, 0xff7280);

    reg32_write(ASRC_ASRPM4, 0xff7280);

    reg32_write(ASRC_ASRPM5, 0xff7280);

    reg32_write(ASRC_ASRQFIFO1,0x001f00);

//  reg32_write(ASRC_ASRMCRA,0x001f00);

//  reg32_write(ASRC_ASRMCRB,0x001f00);

//  reg32_write(ASRC_ASRMCRC,0x001f00);
```

```
}

void sim_ideal_ratio()

 {

   WORD tmp32bit;

   #define ASRSTR_AIDEA_MASK  0x1

   #define ASRSTR_AODFA_MASK  0x1 <<3

   #define ASRSTR_AOLE_MASK   0x1<<6

   #define ASRCTR_DBG_EN              1<<23

   #define ASRCTR_IDRA                1<<13

   #define ASRCTR_USRA                1<<14

   #define ASRC_CLK_PRED_RSTRICTED 0<<28

   #define ASRC_CLK_PRED_DFLT      1<<28 // default: 596MHz div by 2

   #define ASRC_CLK_PRED_DIV3      2<<28 // 596MHz div by 3

   #define ASRC_CLK_PRED_DIV4      3<<28 // 596MHz div by 4

   #define ASRC_CLK_PRED_DIV5      4<<28 // 596MHz div by 5

   #define ASRC_CLK_PRED_DIV6      5<<28 // 596MHz div by 6

   #define ASRC_CLK_PRED_DIV7      6<<28 // 596MHz div by 7

   #define ASRC_CLK_PRED_DIV8      7<<28 // 596MHz div by 8

   #define ECSPI_CLK_PRED_DFLT     1<<25

   #define ECSPI_CLK_PODF_DFLT     1<<19

   #define ASRC_CLK_PODF_DIV1      0<<9  // pred output divide by 1 again

   #define ASRC_CLK_PODF_DIV2      1<<9  // pred output divide by 2 again

   #define ASRC_CLK_PODF_DIV3      2<<9  // pred output divide by 3 again

   #define ASRC_CLK_PODF_DIV4      3<<9  // pred output divide by 4 again

   #define ASRC_CLK_PODF_DFLT      4<<9  // default: pred output divide by 5 again

   #define ASRC_CLK_PODF_DIV6      5<<9  // pred output divide by 6 again

   #define ASRC_CLK_PODF_DIV7      6<<9  // pred output divide by 7 again

   #define ASRC_CLK_PODF_DIV25     24<<9  // pred output divide by 7 again

   #define IEEE_CLK_PRED_DFLT      1<<6  //

   #define IEEE_CLK_PODF_DFLT      4     //

   #define ASR_HFA_HFB         0

   #define ASR_PREMODA_UP2         0<<6

   #define ASR_PREMODA_DIR         1<<6
```

```
    #define ASR_PREMODA_DN2         2<<6

    #define ASR_PREMODA_PAS         3<<6

    #define ASR_POSTMODA_UP2         0<<8

    #define ASR_POSTMODA_DIR         1<<8

    #define ASR_POSTMODA_DN2         2<<8

    // program CCM for ASRC core clocks

     reg32_write(CCM_CCGR7, 0xffffffff); // enable all perihperal clocks during all modes, 
except stop mode

     reg32setbit(CCM_CSCMR2, 21); // Selector for asrc clock multiplexer

     //   0 ipp_asrc_ext

     //   1 pll4_sw_clk(Default): This one is choosen

     reg32_write(CCM_CSCDR2, ASRC_CLK_PRED_DIV8|ECSPI_CLK_PRED_DFLT|ECSPI_CLK_PODF_DFLT|
ASRC_CLK_PODF_DIV25|IEEE_CLK_PRED_DFLT|IEEE_CLK_PODF_DFLT);

     // Disable the ASRC

     reg32_write(ASRC_ASRCTR, 0x0);

     // program AHB clocks

     tmp32bit = reg32_read(CCM_CBCDR);

     tmp32bit = tmp32bit & (~0x00001C00);

     //tmp32bit = tmp32bit | (0x00000C00); // AHB 100MHz // divided-by-4

     //tmp32bit = tmp32bit | (0x00001000); // AHB 80MHz // divided-by-5

     //tmp32bit = tmp32bit | (0x00001400); // AHB 66MHz // divided-by-6

     //tmp32bit = tmp32bit | (0x00001800); // AHB 57MHz // divided-by-7

     tmp32bit = tmp32bit | (0x00001C00); // AHB 50MHz // divided-by-8

     reg32_write(CCM_CBCDR, tmp32bit); // enable all perihperal clocks during all modes, 
except stop mode

     while ( (reg32_read(CCM_CDHIPR) & 0x00008) != 0);

            asrc_config_alloc ( 0x002 | ASRCTR_IDRA | ASRCTR_USRA,    // ASRCTR_VAL, Use 
Ratio input, use ideal ratio, Enable Pair A,

            0x0, //0x09,                                              // ASRIER_VAL, Open 
PairA input and output interrupt

            0x002,     // ASRCNCR_VAL, assign 2 channels to Pair A

            ASR_PREMODA_DIR | ASR_POSTMODA_DIR | ASR_HFA_HFB,         // ASRCFG_VAL, 
POSTMODA=downsampling by 2 ; PREMODA=downsampling by 2

            0x03b03b , // ASRCDR1_VAL, AOCPA=3(FoutA/(2^3)); AICPA=3(FinA/(2^3)); 
AOCDA=7(div 8); AICDA=7(div 8);

            0x0  ,     // ASRCDR2_VAL,
```

```
             0x00d00d   // ASRCSR_VAL,  AOCSA=d: bit clock d: ASRCK1 clk from CCM; AICSA=d: 
bit clock d: ASRCK1 clock from CCM;

             );

      reg32_write(ASRC_ASRIDRHA, 0x04); //

      reg32_write(ASRC_ASRIDRLA, 0x0); // Ideal Ratio is set to be 1.

      #define OUTFIFO_THRESH_0  8<<12

      #define INFIFO_THRESH_1   32

      reg32_write(ASRC_ASRMCRA, OUTFIFO_THRESH_0 | INFIFO_THRESH_1);

      reg32clrbit(ASRC_ASRMCRA, 23);   // zeroize Pair A buffers

      reg32setbit(ASRC_ASRMCRA, 21);   // stall conversion in case of near full/near empty 
condition

      reg32clrbit(ASRC_ASRMCRA, 20); // Do not bypass polyA filter

      // Set ASRC Interrupt

      //CAPTURE_INTERRUPT(ASRC_INT_ROUTINE, asrc_handler);

      //enable_hdler(ASRC_INT_NUM);

      disable_hdler(ASRC_INT_NUM);

      incnt=0;

      outcnt=0;

      reg32setbit(ASRC_ASRCTR,0); // enable ASRC

      #define ASRCFG_INIA_FINISH  0x1<<21

      while ( (reg32_read(ASRC_ASRCFG) & ASRCFG_INIA_FINISH) == 0); // wait for ini finished.

      // Polling

          while (outcnt < 100) <

      {

              int ii;

              if ( (reg32_read(ASRC_ASRSTR) & ASRSTR_AIDEA_MASK) != 0 )

               {

                 for (ii=0;ii<2;ii++)</codeblock

                   {

                      reg32_write(ASRC_ASRDIA,asrc_input_array[incnt]);     // feed in input 
data

                      reg32_write(ASRC_ASRDIA,asrc_input_array[incnt]);     // feed in input 
data

                      incnt=(incnt+1)%128;

                   }
```

```
                }
              if ( (reg32_read(ASRC_ASRSTR) & ASRSTR_AODFA_MASK) != 0 )
               {
                   for (ii=0;ii<2;ii++)<
                     {
                       WORD TempRdOut;
                       TempRdOut=reg32_read(ASRC_ASRDOA);  // get output data
                       TempRdOut=reg32_read(ASRC_ASRDOA);  // get output data
                       outcnt=outcnt+1;
                     }
                 }
              if ( (reg32_read(ASRC_ASRSTR) & ASRSTR_AOLE_MASK) != 0 )
               {
                   reg32_write(ASRC_ASRSTR,ASRSTR_AOLE_MASK);      // clear overloading 
errors
                }
             }
         reg32clrbit(ASRC_ASRCTR,0); // disable ASRC
   }
```

Chapter 54
Sony/Philips Digital Interface (SPDIF)

54.1 Introduction

The Sony/Philips Digital Interface (SPDIF) audio block is a stereo transceiver that allows the processor to receive and transmit digital audio. The SPDIF transceiver allows the handling of both SPDIF channel status (CS) and User (U) data and includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency.

A recovered clock is provided to drive both internal and external components in the system such as ESAI ports, as well as external A/Ds or D/As, with clocking control provided via related registers.

As the SPDIF internal data width is 24-bit, the eight most-significant bits of all registers return zeros.

The figure below shows a block diagram of the SPDIF transceiver data paths (receiver and transmitter) and its interface.

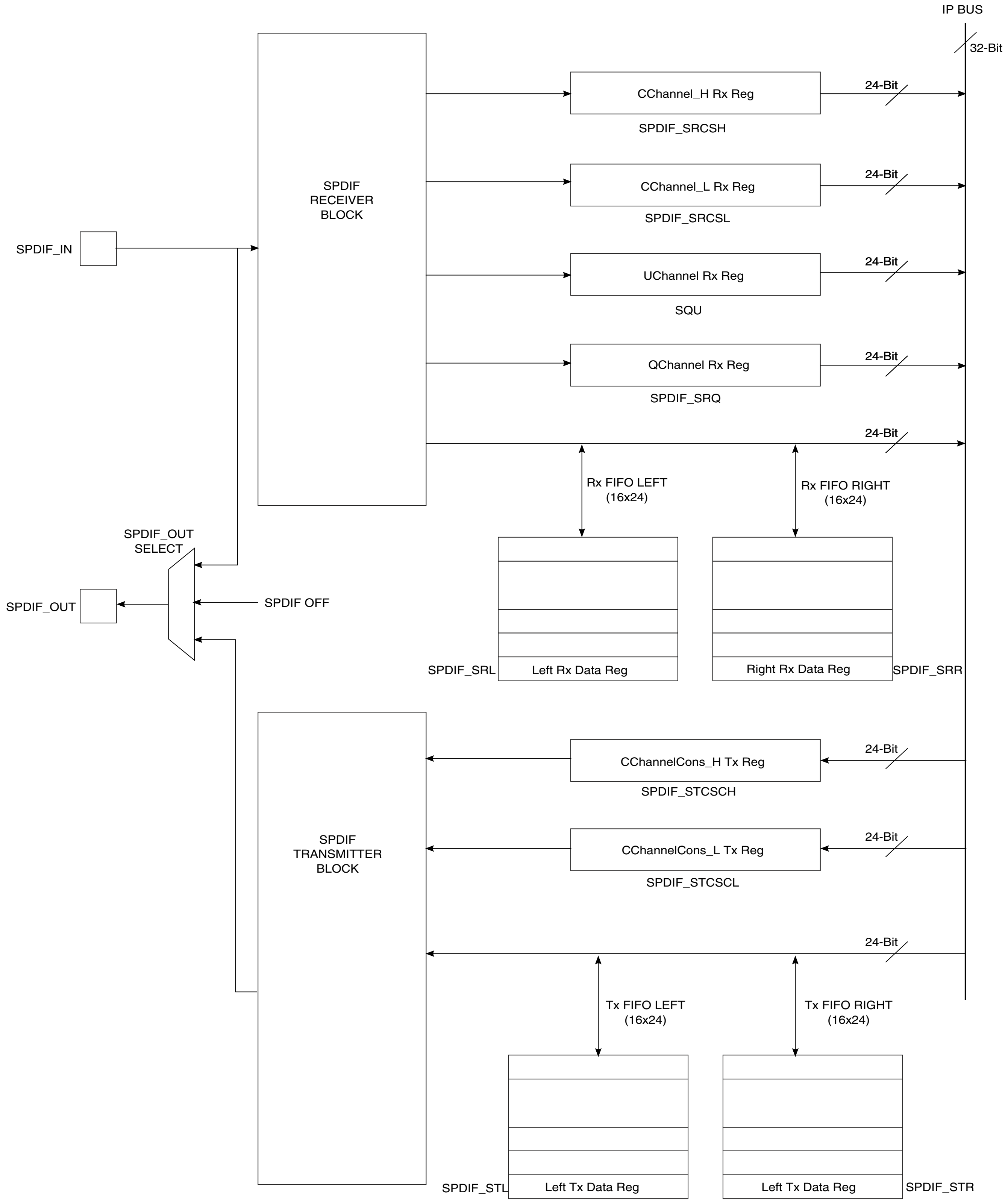


Figure 54-1. SPDIF Transceiver Data Interface Block Diagram

54.1.1 Overview

The SPDIF is composed of two parts: SPDIF Receiver and SPDIF Transmitter.

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in the SPDIF Rx left and right FIFOs. The Channel Status and User Bits are also extracted from each frame and placed in the corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

For the SPDIF transmitter, the audio data is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers. The Channel Status bits are also provided via the corresponding registers. The SPDIF transmitter generates a SPDIF output bitstream in the biphase mark format (IEC60958), which consists of audio data, channel status and user bits.

In the SPDIF transmitter, the IEC60958 biphase bit stream is generated on both edges of the SPDIF Transmit clock. The SPDIF Transmit clock is generated by the SPDIF internal clock generate block and the sources are from outside of the SPDIF block. For the SPDIF receiver, it can recover the SPDIF Rx clock. Both the Rx clock and Tx clock are sent to the ASRC.

54.2 External Signal Description

Table 54-1. Signal Properties

Signal Name	Signal Type	Description
SPDIFIN	input	SPDIF Input Line IEC60958 data in biphase mark format.
SPDIFOUT	output	SPDIF Output Line IEC60958 data in biphase mark format. (Consumer C channel).

54.3 Functional Description

54.3.1 SPDIF Receiver

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in Rx left and right FIFOs.

The Tx left and right FIFOs are 16-deep and 24-wide (equal to the audio data width). The Channel Status and User Bits are also extracted from each frame and placed in corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

The SPDIF receiver handles the main data audio stream and recovers the bit clock from the SPDIF input signal. The sample rate can be determined from the frequency measuring block. Additionally, the receiver supports the SPDIF C and U channels. The SPDIF C and U channel data is interfaced directly to memory-mapped registers.

All the data registers are controlled by the Interrupt Control Block and transferred to the memory-mapped IP bus.

The following functions are performed by the SPDIF receiver:

- Audio Data Reception
- Channel Status bits Reception
- U Channel bits Reception
- Validity Flag Reception
- SPDIF Receiver Exception support
- SPDIF Lock Detection

54.3.1.1 Audio Data Reception

The SPDIF Receiver block extracts the audio data from the IEC60958 stream, and outputs this via Rx left and right FIFOs to the memory-mapped registers SPDIFRxLeft and SPDIFRxRight. Data from the SPDIF receiver is buffered in receive FIFO, and can be read by the processor from the memory-mapped registers.

- **SPDIF receiver data registers - Behavior on overrun, underrun**

The SPDIF Data Receive registers (SPDIFRxLeft and SPDIFRxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect only one part (left or right) of any FIFOs. To prevent this from happening, hardware has been added to the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If a SPDIF Data Rx FIFO overrun occurs on e.g. the right half of the FIFO, the sample that caused the overrun is not written to the right half (due to overrun). Special hardware will make sure the next sample is not written to the left half of the FIFO. If the overrun occurs on the left half of the FIFO, the next sample is not written to the right half of the FIFO.

- **SPDIF receiver data registers - Automatic resynchronization of FIFOs**

An automatic FIFO resynchronization feature is available. It can be enabled and disabled separately for every FIFO. If it is enabled, the hardware will check to see if the left and right FIFOs are in sync. If that is not the case, it will set the filling pointer of the right FIFO to be equal to the filling pointer of the left FIFO.

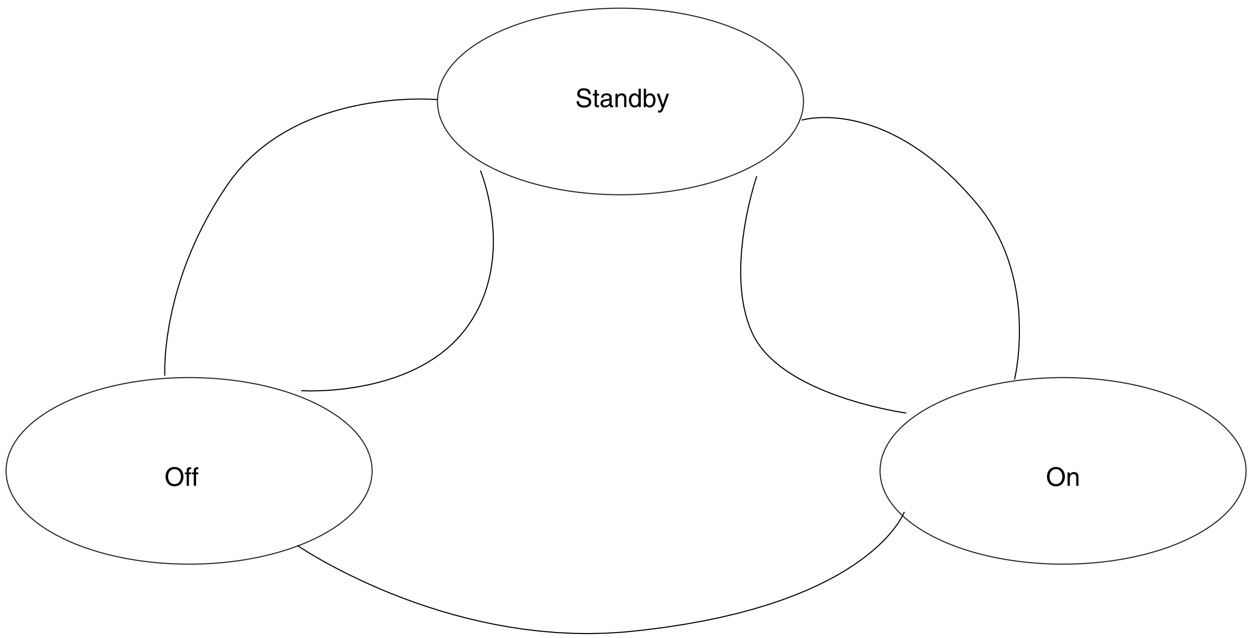


Figure 54-2. FIFO Auto-resync Controller State Machine

The operation is explained from the state diagram shown above. Every FIFO auto-resync controller has a state machine with 3 states: Off, StandBy and On. In the On state, the filling of the left FIFO is compared with the filling of right, and if they are not equal, right is made equal to left, and an interrupt is generated.

The controller will stay in Off state when the feature is disabled. When not disabled, the state machine will go to Off state on any processor read or write to the FIFO. It will go from On or Off to Standby on any left sample read from SPDIF Tx FIFOs, or on any left sample write to SPDIF Rx FIFOs. The controller will go from Standby to On on any right

sample read from SPDIF Tx FIFO, or on any right sample write to SPDIF Rx FIFO. There is a control bit in the SPDIFConfig register to enable/disable the feature for the SPDIF Rx FIFO and SPDIF Tx FIFO.

54.3.1.1.1 Application Note

The automatic FIFO resynchronization can be switched on, and will avoid all mismatches between left and right FIFOs, if the software obeys the following rules:

- When the left data is read or written to the left FIFO, in the same place of the program, data must be read or written to the right FIFO. Maximum time difference between left and right is 1/2 sample clock. (E.g. if sample frequency is 44 Khz, approximately 10 micro-seconds. For 88 Khz, approximately 5 micro-seconds.)
- Write/read data to FIFO s at least 2 samples at the time. If there is a mismatch Left-Right, the resync logic may go on only 1 sample clock after last data is read/written to the FIFO. Also acceptable is polling the FIFO, if at least part of the time 2 samples will be read/written to it.

SPDIF receiver - Additional features

There are three exceptions associated with the SPDIF Receivers FIFOs

- full
- under/overrun
- resync

When the "full" condition is set for processor data input registers, the processor should read data from the FIFO, before overrun occurs. When "full" is set, and the FIFO contains e.g. 6 samples, it is acceptable for the software to read first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 6 samples from the RIGHT address, followed by 6 samples from the LEFT address, or 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times. There is no order specified.

The implementation for SPDIF Rx is a double FIFO, one for left and one for right. "full" is set when both FIFOs are full. "underrun, overrun" are set when one of the FIFOs do underrun or do overrun. The resync interrupt means hardware took special action to resynchronize left and right FIFOs.

The FIFO level at which the "full" interrupt is generated, is programmable via the Full Select field in the SPDIFConfigReg register.

Rx FIFO on and Rx FIFO reset

Two additional control fields of the SPDIF Rx FIFO are the on/off select and FIFO reset fields.

If on/off select is set to off, all-zero will be read from the FIFO, irrespective of the data received over the SPDIF interface.

If FIFO reset is set, the FIFO is blocked at "1 sample in FIFO". In this, the full interrupt will be on if FullSelect is set to "00". If FullSelect is set to any other value, interrupt will be off. The other interrupts are always off.

54.3.1.2 Channel Status Reception

A total of 48 channel status bits are received in two registers. No interpretation is performed by the SPDIF receiver block.

Channel Status Bits are ordered first bit left. CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFRxCChannel_h. CS-channel bit "23" is considered the LSB bit 0 in the register. C-channel bit 24 to 47 is seen as [23:0] bits of register SPDIFRxCChannel_l.

NOTE

Single DMA channel cannot service SPDIF Tx or Rx. Therefore, two DMA channels linked to each another are required to service SPDIF Rx request.

54.3.1.2.1 Channel Status Interrupt

When the value of a new SPDIF "CS" channel status frame is loaded in the register, an interrupt is generated. The interrupt is cleared when the processor writes the corresponding bit in the Interrupt register.

54.3.1.3 User Bit Reception

There are two modes for U Channel reception, CD and non-CD. As is decided by USyncMode (bit 1 of CDText_Control register).

- **Behavior of U Channel receive interface on incoming CD U Channel Sub-code in SPDIF receiver**.

This mode is selected if UsyncMode, bit 1 in register CD Text control is set "1".

The CD sub-code stream embedded into the SPDIF U channel consists of a sequence of packets. Every packet is made up 98 "symbols". The first two symbols of every packet are "sync symbols", the other 96 symbols are "data symbols".

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

Data symbols are coming in MSB first. The MSB is the leading one.

When a "long pause" is seen between 2 subsequent "data symbols", the SPDIF receiver will assume the reception of one or more "sync symbols". Table below gives details.

Table 54-2. Sync Control Bits

Number of U Channel zero bits	Corresponding number of sync symbols
0-1	Unpredictable, not allowed
2-10	0
11-22	1
23-34	2
35-44	3
>45	Unpredictable, not allowed

The recognition of the number of sync symbols derives from the fact that the U channel transmitter in the CD channel decoder will transmit one symbol on average every 12 SPDIF channel bits. On this average rate, there is a maximum tolerance of 5%.

The SPDIF receiver is tolerant of symbol errors. Due to the physical nature of the transmission of the data over the CD disc, not more than 1 out of any 5 consecutive user channel symbols may be in error. The error may cause a change in data value, which is not detected by this interface, or it may cause a data symbol to be seen as a sync symbol, or a sync symbol to be seen as a data symbol. However, not more than 1 out of any 5 consecutive user channel symbols should be affected in this way.

The SPDIF U channel circuitry recognizes the 98-symbol packet structure, and sends the 96 symbol payload to the processor application. The 96 symbol payload is transmitted to the processor via 2 registers:

- The SPDIFRxUChannel register. In this register, data is presented 3 symbols at the time to the processor. Every time 3 new valid symbols, received on the SPDIF U Channel are present, the UChannelRxFull interrupt is asserted. For one 98-symbol packet, 96 symbols are carried across SPDIFRxUChannel. To transfer all this data, 32 UChannelRxFull interrupts are generated.
- The QChannelReceive register. In this register, only the Q bit of the packet is accumulated.Operation is similar to UChannelReceive. Because only Q-bit is transferred, only 96 Q-bits are transferred for any 98-symbol packet. To transfer this data, 4 QChannelRxFull interrupts are generated. When QChannelRxFull occurs, it is coincident with UChannelRxFull. There is only one QChannelRxFull for every 8

UChannelRxFull. The convention is that most significant data is transmitted first, and is left-aligned in the registers.

- Timing regarding packet boundary is extracted by hardware. The last UChannelRxFull corresponding to a given packet should be coincident with the last QChannelRxFull. In this last U, Q channel interrupt, symbols 95-98 are received, Q channel bits 67-98. The interrupts are coincident with UQSyncFound, flagging last symbols of the current frame.
- When the start of the new packet is found before the current packet is complete (less than 98 symbols in the packet), the UQFrameError interrupt is set. The application software should read out UChannelReceive and QchannelReceive registers, discard the value, and assume the start of a new packet.
- As already said, packet sync extraction is tolerant for single-symbol errors. Packet sync detection is based on the recognition of the sequence data-sync-sync-data in the symbol stream, because this is the only syncing sequence that is not affected by single errors. If the sync symbols are not found 98 symbols after the previous occurrence, it is assumed to be destroyed by channel error, and a new sync symbols is interpolated.
- Normally, only data bytes are passed to the application software. Every databyte will have its most significant bit set. If sync symbols are passed to the application software, they are seen as all-zero symbols. Sync symbols can only end up in the data stream due to channel error.

- **Behavior of U Channel receive interface on incoming non-CD data**.

This mode is selected if UsyncMode, bit 1 in register CD Text control is set'0'.

In non-CD mode, the SPDIF U channel stream is recognized as a sequence of "data symbols". No packet recognition is done.

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

3 consecutive data symbols seen in the SPDIF U Channel stream are grouped together into the SPDIFRxUChannel register. First symbol is left, last symbol is right aligned. When SPDIFRxUChannel contains 3 new data symbols, UChannelRxFull is asserted.

In this mode, the operation of QchannelRx and associated interrupt QchannelRxFull is reserved, undefined. And the operation of UQFrameError and UQSyncFound is also reserved, undefined.

The U channel is extracted, and output by the SPDIF Rx on SPDIFRxUChannel-Stream.

When incoming SPDIF data parity error or bit error is detected, and if the next SPDIF word for that channel is error-free, the SPDIF word in error is replaced with the average of the previous word and next word. When incoming SPDIF data parity error or bit error is detected, and the next SPDIF word is in error, the previous SPDIF word is repeated.

54.3.1.4 Validity Flag Reception

An interrupt is associated with the Validity flag. (interrupt 16 - SPDIFValNoGood). This interrupt is set every time a frame is seen on the SPDIF interface with the validity bit set to "invalid".

54.3.1.5 SPDIF Receiver Interrupt Exception Definition

Several SPDIF exceptions can trigger an interrupt.

They are:

- Control Status channel change. Set when SPDIFRxCChannel1 register is updated. The register is updated for every new C-Channel received. The exception is reset on write to InterruptClear register.
- SPDIF Illegal Symbol. Set on reception of illegal symbol during SPDIF receive. Reset by writing register InterruptClear.[1]
- SPDIF bit error. Set on reception of bit error. (Parity bit does not match). Reset on write to InterruptClear register.
- Receive data FIFO full. Set when SPDIF receive data FIFO is full.
- Receive data FIFO underrun/overrun. Set when there is a underrun/overrun on the SPDIF receive data FIFO.
- Receive data FIFO resynchronization. Set when a resynchronization event occurs on the SPDIF receive data FIFO.
- Receive U Channel buffer full. Set when next 24 bits of U channel code are available.
- Receive Q Channel buffer overrun. Set when Q channel buffer overrun.
- Receive U Channel buffer overrun. Set on U channel buffer overrun.
- Receive Q Channel buffer full. Set when next 24 bits of Q channel code are available.

1. The SPDIF input is a biphase/mark modulated signal. The time between any two successive transitions of the SPDIF signal is always 1, 2 or 3 SPDIF symbol periods long. The SPDIF receiver will parse the stream, and split it in so-called symbols. It recognizes s1, s2 and s3 symbols, depending on the length of the symbols. Not all sequences of these symbols are allowed. To give an example, a sequence s2-s1-s1-s1-s2 cannot occur in a no-error SPDIF signal. If the receiver finds such an illegal sequence, the illegal symbol interrupt is set. No corrective action is undertaken. When the interrupt occurs, this means that(a) The SPDIF signal is destroyed by noise (b) The SPDIF frequency changed.

- Receive UQ sync found. Set when UQ channel sync found.
- Receive UQ frame error. Set when UQ frame error found.

54.3.1.6 Standards Compliance

The SPDIF interface is compatible with the Tech 3250-E standard of the European Broadcasting Union, except clause 6.3.3 and the IEC60958-3 Ed2 for relevant topics.

Supported input frequency range is 12 Khz up to 96 Khz. (fully compliant) and 96 Khz up to 176 Khz (Can interface with compliant SPDIF transmitter within same cabinet, making reasonable assumptions on jitter added due to interconnecting wire.)

Tolerated jitter on SPDIF input signals are 0.25 bit peak-peak for high frequencies. There is no jitter limit for low frequencies. The user channel extraction in CD mode is capable of coping with single-symbol errors, and still retrieve U channel frames on correct boundaries. This capability is required for reliable reception of CD-Text from some Philips CD channel decoders. This capability was deemed more important than compliance with the IEC60958 annex A.3 standard, and for this reason user channel reception is not compliant with IEC60958 annex A.3. However, the interface is capable to receive U channel inserted by a typical CD channel decoder. Also, in this case, it is more robust and tolerant for channel error than what is required by IEC60958 annex A.3.

54.3.1.7 SPDIF PLOCK Detection and Rxclk Output

Using the high speed system clock, the internal DPLL can extract the bit clock (advanced pulse) from the input bitstream. When this internal DPLL is locked, the LOCK bit of PhaseConfig Register will be set, and the SPDIF Lock output pin PLOCK will be asserted.

After DPLL has locked, the pulses are generated, and the average pulse rate is 128 x the sampling frequency.(For a 44.1 Khz input sampling frequency, the average pulse rate = 128 x 44.1 Khz.) The pulse signal is used in the FreqMeas circuit to generate the frequency measurement result.

54.3.1.8 Measuring Frequency of SPDIF_RxClk

The internal DPLL can extract the bit clock (advanced plus) from the input bitstream. To do that, it is necessary to measure the frequency of the incoming signal in relationship with the system clock (BUS_CLK).

Associated with it are two registers, PhaseConfig and FreqMeas. The circuit will measure the frequency of the incoming clock as a function of the BUS_CLK. The circuit is a second-order filter. The output is a value represented by an unsigned number stored in the 24-bit FreqMeas register, giving the frequency of the source as a function of the BUS_CLK.

FreqMeas[23:0] = FreqMeas_CLK / BUS_CLK * 2^{10} * GAIN.

For example, if the GAIN is selected as 8*210 (PhaseConfig[5:3] = 3'b011), the actual result

FreqMeas_CLK / BUS_CLK is equal to FreqMeas[23:0] / 2^{23}.

54.3.2 SPDIF Transmitter

Audio data for the SPDIF transmitter is provided by processor via the SPDIFTxLeft and SPDIFTxRight registers.

Clocking for SPDIF transmitter is from either the osc_audio, ipg_baud_spdif_clk, hckt, or spdif_extclk, or frequency divided ipg_clk. A multiplexer is used to choose the clock source. The SPDIF transmitter clock source can be divided down as needed using Txclk_DF. The SPDIF transmitter output can be chosen from either the SPDIF transmitter block, directly from the SPDIF receiver (via the output multiplexer), or disabled.

The SPDIF transmitter generates a SPDIF output bitstream in IEC60958 biphase mark format, consisting of audio data, channel status.

54.3.2.1 Audio Data Transmission

Audio data for the SPDIF transmitter is provided by the processor via SPDIFTxLeft and SPDIFTxRight registers. They send audio data to Tx left and right FIFOs. The Tx left and right FIFOs are also 16-deep and 24-width (equal to the audio data width).

- **SPDIF transmitter data registers - Behavior on overrun, underrun**

The SPDIF Data Transmit registers (SPDIFTxLeft and SPDIFTxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect only one part (left or right) of any FIFO. To prevent this from happening, hardware has been added on the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If SPDIF Tx FIFO underruns on the right half of the FIFO, no sample leaves that FIFO (because it was already empty). Special hardware will make sure that the next sample read from the left FIFO will not leave the FIFO (no read strobe is generated). If the underrun occurs on the left half of the FIFO, next read strobe to the right FIFO is blocked.

- **SPDIF transmitter data registers - Automatic resynchronization of FIFOs**

See Audio Data Reception.

- **SPDIFTxLeft, SPDIFTxRight details**

With SPDIF Tx FIFOs three exceptions are associated.

- empty
- under/overrun
- resync

When the empty condition is set for processor data output registers, the processor should write data to the FIFO, before underrun occurs. When empty is set and, for instance, 6 samples need to be written, it is acceptable for the software to write first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times. Left should be written before right. The implementation of all data out FIFOs is a double FIFO, one for left and one for right. Empty is set when both FIFOs are empty. Underrun, overrun are set when one of the FIFOs do underrun or do overrun. Resync is set when the hardware resynchronizes left and right FIFOs.

On receiving underrun, overrun interrupt, synchronization between Left and Right words in the FIFOs may be lost. Synchronization will not be lost when the underrun or overrun comes from the IEC60958 side of the FIFO. If the processor reads or writes more data from, for example, left than from right, synchronization will be lost. If automatic resynchronization is enabled, and if the software obeys the rules to let this work, resynchronization will be automatic.

54.3.2.2 Channel Status Transmission

A total of 48 Consumer channel status bits are transmitted from two registers. Channel Status Bits are ordered first bit left.

CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFTxCChannelCons_h. CS-channel bit "23" is considered bit 0 in the register. C-channel bits 24-47 are seen as MSB-LSB bits of register SPDIFTxCChannelCons_l.

54.3.2.3 Validity Flag Transmission

The validity bit setting is performed via bit 5 of the SPDIF_SCR register.

54.4 Programmable Registers

SPDIF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_1000	SPDIF Configuration Register (SPDIF_SCR)	32	R/W	0000_0400h	54.4.1/3017
4006_1004	CDText Control Register (SPDIF_SRCD)	32	R/W	0000_0000h	54.4.2/3019
4006_1008	PhaseConfig Register (SPDIF_SRPC)	32	R/W	0000_0000h	54.4.3/3020
4006_100C	InterruptEn Register (SPDIF_SIE)	32	R/W	0000_0000h	54.4.4/3022
4006_1010	InterruptStat Register (SPDIF_SIS)	32	R	0000_0000h	54.4.5/3025
4006_1010	InterruptClear Register (SPDIF_SIC)	32	W	0000_0000h	54.4.6/3028
4006_1014	SPDIFRxLeft Register (SPDIF_SRL)	32	R	0000_0000h	54.4.7/3030
4006_1018	SPDIFRxRight Register (SPDIF_SRR)	32	R	0000_0000h	54.4.8/3030
4006_101C	SPDIFRxCChannel_h Register (SPDIF_SRCSH)	32	R	0000_0000h	54.4.9/3031
4006_1020	SPDIFRxCChannel_l Register (SPDIF_SRCSL)	32	R	0000_0000h	54.4.10/ 3031
4006_1024	UchannelRx Register (SPDIF_SRU)	32	R	0000_0000h	54.4.11/ 3032
4006_1028	QchannelRx Register (SPDIF_SRQ)	32	R	0000_0000h	54.4.12/ 3032
4006_102C	SPDIFTxLeft Register (SPDIF_STL)	32	W	0000_0000h	54.4.13/ 3033
4006_1030	SPDIFTxRight Register (SPDIF_STR)	32	W	0000_0000h	54.4.14/ 3033
4006_1034	SPDIFTxCChannelCons_h Register (SPDIF_STCSCH)	32	R/W	0000_0000h	54.4.15/ 3034
4006_1038	SPDIFTxCChannelCons_l Register (SPDIF_STCSCL)	32	R/W	0000_0000h	54.4.16/ 3034
4006_1044	FreqMeas Register (SPDIF_SRFM)	32	R	0000_0000h	54.4.17/ 3035
4006_1050	SPDIFTxClk Register (SPDIF_STC)	32	R/W	0002_0F00h	54.4.18/ 3035

54.4.1 SPDIF Configuration Register (SPDIF_SCR)

Address: 4006_1000h base + 0h offset = 4006_1000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved								RxFIFO_Ctrl	RxFIFO_En	RxFIFO_Rst	RxFIFOFull_Sel		RxAutoSync	TxAutoSync	TxFIFOEmpty_Sel
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	TxFIFOEmpty_Sel	Reserved	Low_Power	Soft_Reset	TxFIFO_Ctrl		DMA_Rx_En	DMA_TX_En	Reserved		ValCtrl	TxSel			USrc_Sel	
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

SPDIF_SCR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 RxFIFO_Ctrl	Controls the values read from Rx data Register 0 Normal operation 1 Always read zero from Rx data register
22 RxFIFO_En	Enables/Disables FIFO accepting data from the interface 0 SPDIF Rx FIFO is on 1 SPDIF Rx FIFO is off. Does not accept data from interface
21 RxFIFO_Rst	Defines the reset state of RxFIFO 0 Normal operation 1 Reset register to 1 sample remaining
20–19 RxFIFOFull_Sel	Defines the threshold for generation of FIFO Full interrupt 00 Full interrupt if at least 1 sample in Rx left and right FIFOs 01 Full interrupt if at least 4 sample in Rx left and right FIFOs

Table continues on the next page...

SPDIF_SCR field descriptions (continued)

Field	Description
	10 Full interrupt if at least 8 sample in Rx left and right FIFOs 11 Full interrupt if at least 16 sample in Rx left and right FIFO
18 RxAutoSync	Controls automatic synchronization of FIFO's. If enabled will check if the left and right Rx FIFOs are in sync. 0 Rx FIFO auto sync off 1 RxFIFO auto sync on
17 TxAutoSync	Controls automatic synchronization of FIFO's. If enabled will check if the left and right Tx FIFOs are in sync. 0 Tx FIFO auto sync off 1 Tx FIFO auto sync on
16–15 TxFIFOEmpty_Sel	Defines the threshold for generation of empty interrupt 00 Empty interrupt if 0 sample in Tx left and right FIFOs 01 Empty interrupt if at most 4 sample in Tx left and right FIFOs 10 Empty interrupt if at most 8 sample in Tx left and right FIFOs 11 Empty interrupt if at most 12 sample in Tx left and right FIFOs
14 -	This field is reserved. Reserved
13 Low_Power	When write 1 to this bit, it will cause SPDIF enter Low_Power mode. return 1 when SPDIF in Low_Power mode.
12 Soft_Reset	When write 1 to this bit, it will cause SPDIF software reset. The software reset will last 8 cycles. When in the reset process, return 1 when read. else return 0 when read.
11–10 TxFIFO_Ctrl	Controls the Transmit operation from the Tx FIFO 00 Send out digital zero on SPDIF Tx 01 Tx Normal operation 10 Reset to 1 sample remaining 11 Reserved
9 DMA_Rx_En	DMA Receive Request Enable (RX FIFO full)
8 DMA_TX_En	DMA Transmit Request Enable (Tx FIFO empty)
7–6 -	This field is reserved. Reserved
5 ValCtrl	Defines the validity control of data. 0 Outgoing Validity always set 1 Outgoing Validity always clear
4–2 TxSel	Transmit channel select 000 Off and output 0

Table continues on the next page...

SPDIF_SCR field descriptions (continued)

Field	Description
	001 Feed-through SPDIFIN 101 Tx Normal operation
1–0 USrc_Sel	Defines the source of U Channel. 00 No embedded U channel 01 U channel from SPDIF receive block (CD mode) 10 Reserved 11 U channel from on chip transmitter

54.4.2 CDText Control Register (SPDIF_SRCD)

Address: 4006_1000h base + 4h offset = 4006_1004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	Reserved							0					Reserved	USyncMode	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_SRCD field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–15 Reserved	This read-only field is reserved and always has the value 0.
14–8 -	This field is reserved. Reserved

Table continues on the next page...

SPDIF_SRCD field descriptions (continued)

Field	Description
7–3 Reserved	This read-only field is reserved and always has the value 0.
2 -	This field is reserved. Reserved
1 USyncMode	Defines the Mode for User channel reception (CD and non CD) 0 Non-CD data 1 CD user channel subcode
0 -	This field is reserved. Reserved

54.4.3 PhaseConfig Register (SPDIF_SRPC)

Address: 4006_1000h base + 8h offset = 4006_1008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					ClkSrc_Sel				LOCK	GainSel			0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_SRPC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.

Table continues on the next page...

SPDIF_SRPC field descriptions (continued)

Field	Description
23–11 Reserved	This read-only field is reserved and always has the value 0.
10–7 ClkSrc_Sel	Clock source selection, all other settings not shown are reserved: **NOTE:** For the ClkSrc_Sel, each of the input clocks has the frequency of 64*Audio_Sample_Rate. And the Audio_Sample_Rate can be 32 KHz to 192 KHz. 0000 if (DPLL Locked) SPDIF_RxClk else extal_clk 0001 if (DPLL Locked) SPDIF_RxClk else pll4_div_clk 0010 if (DPLL Locked) SPDIF_RxClk else External audio clock input 0011 if (DPLL Locked) SPDIF_RxClk else mlb_clk 0100 if (DPLL Locked) SPDIF_Rxclk else esai_hckt 0101 extal_clk 0110 pll4_div_clk 0111 external audio clok 1000 mlb_clk 1001 esai_hckt 1010 if (DPLL Locked) SPDIF_RxClk else SAI0_TXBCLK 1011 if (DPLL Locked) SPDIF_RxClk else SAI3_TXBCLK 1100 sai0_txbclk 1101 sai3_txbclk
6 LOCK	LOCK bit to show that the internal DPLL is locked, read only
5–3 GainSel	Gain selection: 000 24*2**10 001 16*2**10 010 12*2**10 011 8*2**10 100 6*2**10 101 4*2**10 110 3*2**10
2–0 Reserved	This read-only field is reserved and always has the value 0.

54.4.4 InterruptEn Register (SPDIF_SIE)

The InterruptEn register (SPDIF_SIE) provides control over the enabling of interrupts.

Address: 4006_1000h base + Ch offset = 4006_100Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								0	Reserved		Lock	TxUnOv	TxResyn	CNew	ValNoGood
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SymErr	BitErr	Reserved			URxFul	URxOv	QRxFul	QRxOv	UQSync	UQErr	RxFIFOUnOv	RxFIFOResyn	LockLoss	TxEm	RxFIFOFul
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_SIE field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 Reserved	This read-only field is reserved and always has the value 0.
22–21 -	This field is reserved. Reserved, for InterruptStat/Clear return zeros when read, for InterruptEn, bit 23 also read zero
20 Lock	SPDIF receiver's DPLL is locked 0 Interrupt disabled 1 Interrupt enabled
19 TxUnOv	SPDIF Tx FIFO under/overrun 0 Interrupt disabled 1 Interrupt enabled
18 TxResyn	SPDIF Tx FIFO resync 0 Interrupt disabled 1 Interrupt enabled
17 CNew	SPDIF receive change in value of control channel 0 Interrupt disabled 1 Interrupt enabled

Table continues on the next page...

SPDIF_SIE field descriptions (continued)

Field	Description
16 ValNoGood	SPDIF validity flag no good 0 Interrupt disabled 1 Interrupt enabled
15 SymErr	SPDIF receiver found illegal symbol 0 Interrupt disabled 1 Interrupt enabled
14 BitErr	SPDIF receiver found parity bit error 0 Interrupt disabled 1 Interrupt disabled
13–11 -	This field is reserved. Reserved. Return zeros when read
10 URxFul	U Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from U Rx reg. 0 Interrupt disabled 1 Interrupt enabled
9 URxOv	U Channel receive register overrun 0 Interrupt disabled 1 Interrupt enabled
8 QRxFul	Q Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from Q Rx reg. 0 Interrupt disabled 1 Interrupt enabled
7 QRxOv	Q Channel receive register overrun 0 Interrupt disabled 1 Interrupt enabled
6 UQSync	U/Q Channel sync found 0 Interrupt disabled 1 Interrupt enabled
5 UQErr	U/Q Channel framing error 0 Interrupt disabled 1 Interrupt enabled
4 RxFIFOUnOv	Rx FIFO underrun/overrun 0 Interrupt disabled 1 Interrupt enabled
3 RxFIFOResyn	Rx FIFO resync 0 Interrupt disabled 1 Interrupt enabled
2 LockLoss	SPDIF receiver loss of lock

Table continues on the next page...

SPDIF_SIE field descriptions (continued)

Field	Description
	0 Interrupt disabled 1 Interrupt enabled
1 TxEm	SPDIF Tx FIFO empty, can't be cleared with reg. IntClear. To clear it, write toTx FIFO. 0 Interrupt disabled 1 Interrupt enabled
0 RxFIFOFul	SPDIF Rx FIFO full, can't be cleared with reg. IntClear. To clear it, read from Rx FIFO. 0 Interrupt disabled 1 Interrupt enabled

54.4.5 InterruptStat Register (SPDIF_SIS)

The InterruptStat (SPDIF_SIS) register is a read only register that provides the status on interrupt operations.

Address: 4006_1000h base + 10h offset = 4006_1010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved											Lock	TxUnOv	TxResyn	CNew	ValNoGood
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SymErr	BitErr	Reserved				URxOv	Reserved	QRxOv	UQSync	UQErr	RxFIFOUnOv	RxFIFOResyn	LockLoss	TxFIFOEmpty	RxFIFOFul
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_SIS field descriptions

Field	Description
31–21 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error

Table continues on the next page...

SPDIF_SIS field descriptions (continued)

Field	Description
13–10 -	This field is reserved. Reserved.
9 URxOv	U Channel receive register overrun
8 -	This field is reserved. Reserved
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
1 TxFIFOEmpty	Tx FIFO Empty Sets the Tx FIFO Empty flag.
0 RxFIFOFul	Rx FIFO full flag Sets the Rx FIFO full flag.

54.4.6 InterruptClear Register (SPDIF_SIC)

The InterruptClear (SPDIF_SIC) register is a write only register and is used to clear interrupts.

Address: 4006_1000h base + 10h offset = 4006_1010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W												Lock	TxUnOv	TxResyn	CNew	ValNoGood
Reset												0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R			Reserved					Reserved							Reserved	
W	SymErr	BitErr					URxOv		QRxOv	UQSync	UQErr	RxFIFOUnOv	RxFIFOResyn	LockLoss		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_SIC field descriptions

Field	Description
31–21 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–10 -	This field is reserved. Reserved.
9 URxOv	U Channel receive register overrun
8 -	This field is reserved. Reserved
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
1–0 -	This field is reserved. Reserved.

54.4.7 SPDIFRxLeft Register (SPDIF_SRL)

SPDIFRxLeft register is an audio data reception register.

Address: 4006_1000h base + 14h offset = 4006_1014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RxDataLeft																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_SRL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 RxDataLeft	Processor receive SPDIF data left

54.4.8 SPDIFRxRight Register (SPDIF_SRR)

SPDIFRxRight register is an audio data reception register.

Address: 4006_1000h base + 18h offset = 4006_1018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RxDataRight																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_SRR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 RxDataRight	Processor receive SPDIF data right

54.4.9 SPDIFRxCChannel_h Register (SPDIF_SRCSH)

SPDIFRxCChannel_h register is a channel status reception register.

Address: 4006_1000h base + 1Ch offset = 4006_101Ch

Bit	31–24	23–0
R	Reserved	RxCChannel_h
W		
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

SPDIF_SRCSH field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 RxCChannel_h	SPDIF receive C channel register, contains first 24 bits of C channel without interpretation

54.4.10 SPDIFRxCChannel_l Register (SPDIF_SRCSL)

SPDIFRxCChannel_l register is a channel status reception register.

Address: 4006_1000h base + 20h offset = 4006_1020h

Bit	31–24	23–0
R	Reserved	RxCChannel_l
W		
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

SPDIF_SRCSL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 RxCChannel_l	SPDIF receive C channel register, contains next 24 bits of C channel without interpretation

54.4.11 UchannelRx Register (SPDIF_SRU)

UChannelRx register is a user bits reception register.

Address: 4006_1000h base + 24h offset = 4006_1024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RxUChannel																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_SRU field descriptions

Field	Description
31–24 [unimplemented]	This field is reserved. This is a 24-bit register the upper byte is unimplemented.
23–0 RxUChannel	SPDIF receive U channel register, contains next 3 U channel bytes

54.4.12 QchannelRx Register (SPDIF_SRQ)

QChannelRx register is a user bits reception register.

Address: 4006_1000h base + 28h offset = 4006_1028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RxQChannel																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_SRQ field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 RxQChannel	SPDIF receive Q channel register, contains next 3 Q channel bytes

54.4.13 SPDIFTxLeft Register (SPDIF_STL)

SPDIFTxLeft register is an audio data transmission register.

Address: 4006_1000h base + 2Ch offset = 4006_102Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																															
W									TxDataLeft																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_STL field descriptions

Field	Description
31–24 [unimplemented]	This field is reserved. This is a 24-bit register the upper byte is unimplemented.
23–0 TxDataLeft	SPDIF transmit left channel data. It is write-only, and always returns zeros when read

54.4.14 SPDIFTxRight Register (SPDIF_STR)

SPDIFTxRight register is an audio data transmission register.

Address: 4006_1000h base + 30h offset = 4006_1030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																															
W									TxDataRight																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_STR field descriptions

Field	Description
31–24 [unimplemented]	This field is reserved. This is a 24-bit register the upper byte is unimplemented.
23–0 TxDataRight	SPDIF transmit right channel data. It is write-only, and always returns zeros when read

54.4.15 SPDIFTxCChannelCons_h Register (SPDIF_STCSCH)

SPDIFTxCChannelCons_h register is a channel status transmission register.

Address: 4006_1000h base + 34h offset = 4006_1034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TxCChannelCons_h																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_STCSCH field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 TxCChannelCons_h	SPDIF transmit Cons. C channel data, contains first 24 bits without interpretation. When read, it returns the latest data written by the processor

54.4.16 SPDIFTxCChannelCons_l Register (SPDIF_STCSCL)

SPDIFTxCChannelCons_l register is a channel status transmission register.

Address: 4006_1000h base + 38h offset = 4006_1038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TxCChannelCons_l																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIF_STCSCL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 TxCChannelCons_l	SPDIF transmit Cons. C channel data, contains next 24 bits without interpretation. When read, it returns the latest data written by the processor

54.4.17 FreqMeas Register (SPDIF_SRFM)

Address: 4006_1000h base + 44h offset = 4006_1044h

Bit	31–24	23–0
R	Reserved	FreqMeas
W	Reserved	
Reset	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

SPDIF_SRFM field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–0 FreqMeas	Frequency measurement data

54.4.18 SPDIFTxClk Register (SPDIF_STC)

The SPDIFTxClk Control register includes the means to select the transmit clock and frequency division.

Address: 4006_1000h base + 50h offset = 4006_1050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								0				SYSCLK_DF			
W	Reserved												SYSCLK_DF			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	SYSCLK_DF					TxClk_Source			tx_all_clk_en	TxClk_DF						
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

SPDIF_STC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–11 SYSCLK_DF	system clock divider factor, 2~512. 0 no clock signal 1 divider factor is 2 511 divider factor is 512
10–8 TxClk_Source	Defines the transmit clock source 000 SPDIF EXTAL (EXTAL_CLK from PAD 79 ALT mode 5) 001 PLL4 DIV CLK 010 AUDIO EXTAL (AUD_EXT_CLK from PAD3 ALT2/PAD5 ALT2 /PAD40 ALT2) 011 MLB CLK IN (MLB CLOCK from PAD1 ALT7/PAD54 ALT6) 100 ESAI HCKT (ESAI High speed Transmitter Clock from PAD78 ALT3) 101 SYS CLK (SPDIF BUS CLOCK) 110 SAI0 TX BCLK (SAI0TX_BLCK Clock PAD93 ALT1) 111 SAI3 TX BCLK (SAI3 TX_BCLK clock PAD16 ALT2)
7 tx_all_clk_en	Spdif transfer clock enable.When data is going to be transfered, this bit should be set to1. 0 disable transfer clock. 1 enable transfer clock.
6–0 TxClk_DF	Divider factor (1-128) 0 divider factor is 1 1 divider factor is 2 127 divider factor is 128

Chapter 55
Display Control Unit (DCU4) (2D-ACE Functionality)

55.1 Display Control Unit (DCU)

55.2 Introduction

The Display Controller Unit (DCU4) module (2D-ACE functionality) is a system master that fetches graphics stored in internal or external memory and displays them on a TFT LCD panel. A wide range of panel sizes is supported and the timing of the interface signals is highly configurable. Graphics are read directly from memory and then blended in real-time, which allows for dynamic content creation with minimal CPU intervention. Graphics may be encoded in a variety of formats to optimise memory usage. The DCU4 also has the capability of displaying real-time video from an external video source.

55.2.1 Overview

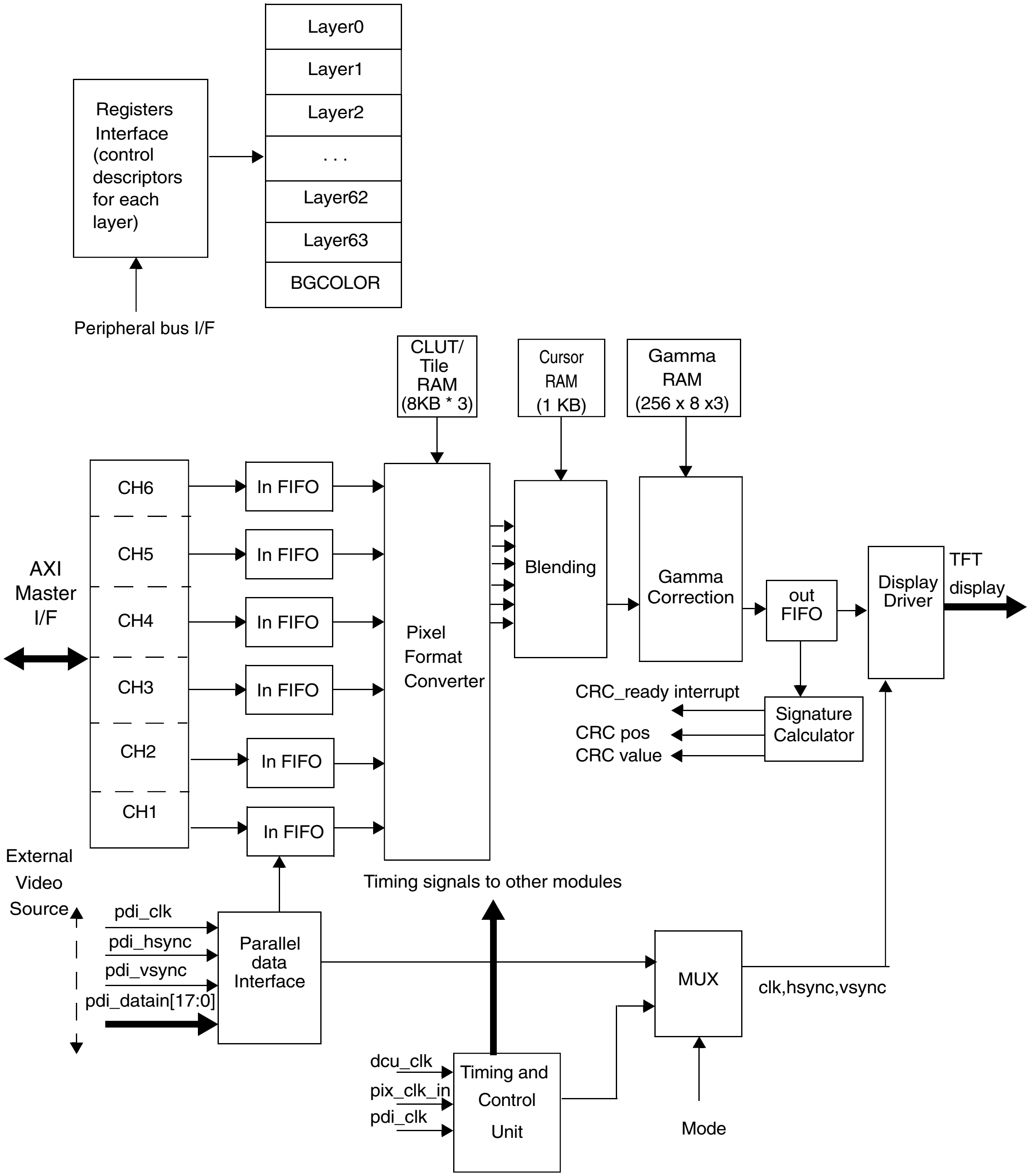


Figure 55-1. Display control unit block diagram

Figure 55-1 shows the DCU4 architecture. This comprises two distinct sections. The lower section shows the functional blocks of the DCU4 that fetch the graphic and video content and drive the TFT LCD panel. The upper section describes the user interface through which the user configures the graphical content of the TFT LCD panel.

The sections are analogous to the structure of communications modules, such as the FlexCAN, where one part of the module is configured to connect with the communications bus through bit-timing, parity, baud rate, etc., while a different part is used to store the data content and message identifiers.

The configuration of the lower section is dependent on the specific TFT LCD panel that is attached to the DCU4 outputs. In most cases, this is configured once for the hardware in use before the DCU4 is enabled. When active, this section automatically:

- Calculates the relevant graphical content for each pixel
- Fetches the source graphics from memory using its internal DMA channels (labelled CH1 to CH6) and stores the data in dedicated FIFO buffers
- Converts the graphic value of each fetched pixel into full quality color format (if required)
- Calculates the required pixel value by blending the values of up to six separate graphics
- Performs a gamma correction on the pixel value (if required)
- Sends the pixel value to the TFT LCD display over its data bus
- Sets flags to indicate end of frame, FIFO buffer threshold, and other status changes

The upper section describes the characteristics of the graphics to be displayed on the panel and how they are blended together. The DCU4 manages the graphical content of the panel through sets of registers called layers. There are 64 layers available in the DCU4 and each contains the following information:

- Horizontal and vertical size of graphic
- Position of graphic on the panel
- Address of graphic in memory
- Color encoding format and color palettes (if required)
- Type and depth of blending
- Range of colors identified for chroma blending
- Tile size
- Foreground and background colors for transparency encoded graphics

The values in these registers may be changed at any time, and the panel content will be updated either under software control or at an automatic point in the panel refresh cycle. The layers are set to a fixed priority, and this is used by the lower section to define which layers are blended, in which order, on the panel.

The upper section also contains configuration registers for a cursor graphic, the default background color, interrupt enables, test graphic, and simple register protection settings.

55.2.2 Features

The DCU4 has these features:

- Full RGB888 output to TFT LCD panel
- 64 graphics layers, a default background color layer and a cursor layer with integrated blinking option
- Blending of each pixel using up to 6 source layers dependent on size of panel
- Programmable panel size up to a maximum of XGA (1024x768) and a typical operating configuration of SVGA (800 x 600)
- Gamma correction with 8-bit resolution on each color component
- Safety mode for tagging pixels on highest priority layers
- Dedicated memory blocks to store a cursor and Color Look Up Tables (CLUTs)
- Temporal Dithering.

Each graphic layer has the following attributes:

- Can be placed with one pixel resolution in either axis
- Can also be placed in negative X and Y directions
- Supports multiple color-encoding formats including:
  - 1, 2, 4 and 8 bits per pixel indexed colors with alpha channel in look-up table
  - APAL8 indexed colors with alpha channel in each pixel
  - RGB565 and RGB888 direct colors without alpha channel
  - ARGB1555, ARGB4444, and BGRA8888 direct colors with an alpha channel
  - YCbCr422 format
- Alpha blending with 8-bit resolution
- Chroma-key blending for anti-mask encoding
- Multiple combined alpha and chroma-key blending modes
- Transparency modes for anti-aliased text and graphics
- Luminance mode for highlighting content
- Tile mode for efficient creation of textured background content

- Support for Run Length Encoding (RLE) compression
- Optimized mode for use with DDR memory

55.2.3 Modes of Operation

The DCU4 has four modes of operation:

- **DCU_OFF**: When in this mode, the DCU4 is turned off. All the logic in the design is put in reset state to reduce power.
- **NORMAL_MODE**: The DCU4 displays and blends the graphics specified by the layer descriptors.
- **PDI_MODE:** A mode which fetches video from an external video source and combines that with the graphics configured on the layers.
- **COLBAR_MODE**: Color bar generation for testing purposes.

55.3 External Signal Description

This section covers the DCU interface signals.

55.3.1 Overview

The choice of signals used depends on the configuration of the DCU4. All active signals must be selected by configuring the appropriate IOMUX registers.

The DCU4 has up to 22 input signals and up to 30 output signals, as shown in the following figure. The choice of signals used depends on the configuration of the DCU4. All active signals must be enabled by configuring the appropriate IOMUX registers.

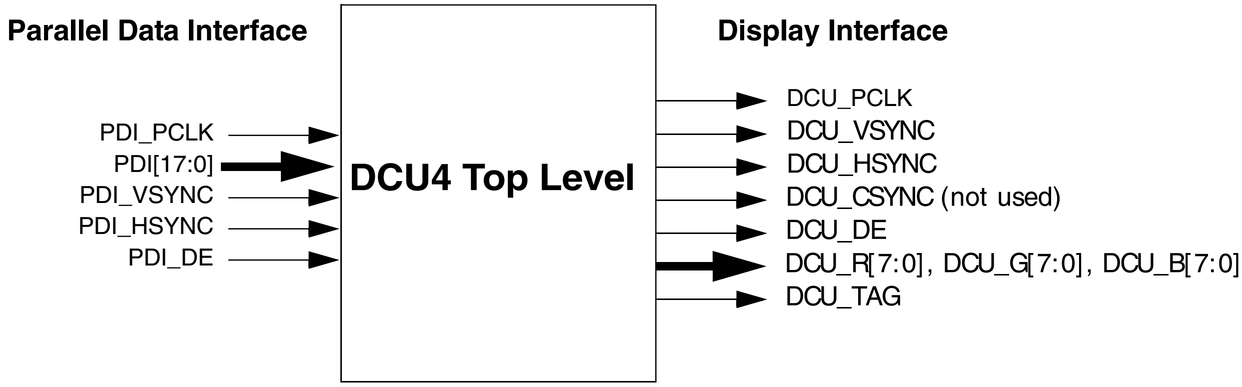


Figure 55-2. External Signals

55.3.2 Detailed Signal Descriptions

Table 55-1. Detailed Signal Descriptions

Signal	Direction	Description
Display Interface		
DCU_PCLK	OUT	Pixel clock used to drive the display panel
DCU_VSYNC	OUT	Vertical sync signal, indicating the beginning of a new frame
DCU_HSYNC	OUT	Horizontal sync signal, indicating the beginning of a new line
DCU_TAG	OUT	When high, this signal indicates that the pixel is tagged and an application can calculate CRC externally on this pixel.
DCU_DE	OUT	Data Enable. Qualifies the data output (dcu_ld)
DCU_R[7:0] DCU_G[7:0] DCU_B[7:0]	OUT	Red, green and blue data output.

55.4 DCU4 Memory Map

Table 55-2. DCU4 memory map

Parameter	Address Range
Register address space	0x0000 – 0x1FFF

Table continues on the next page...

Table 55-2. DCU4 memory map (continued)

Parameter	Address Range
Palette/Tile address space	0x2000 – 0x3FFF
Gamma_R address space	0x4000 – 0x43FF
Gamma_G address space	0x4400 – 0x47FF
Gamma_B address space	0x4800 – 0x4BFF
Cursor address space	0x4C00 – 0x4FFF

55.5 Memory Map and Registers

DCU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8000	Control Descriptor Cursor 1 Register (DCU0_CTRLDESCCURSOR1)	32	R/W	0000_0000h	55.5.1/3101
4005_8004	Control Descriptor Cursor 2 Register (DCU0_CTRLDESCCURSOR2)	32	R/W	0000_0000h	55.5.2/3101
4005_8008	Control Descriptor Cursor 3 Register (DCU0_CTRLDESCCURSOR3)	32	R/W	0000_0000h	55.5.3/3102
4005_800C	Control Descriptor Cursor 4 Register (DCU0_CTRLDESCCURSOR4)	32	R/W	0000_0000h	55.5.4/3103
4005_8010	DCU4 Mode Register (DCU0_DCU_MODE)	32	R/W	0000_8000h	55.5.5/3104
4005_8014	Background Register (DCU0_BGND)	32	R/W	0000_0000h	55.5.6/3107
4005_8018	Display Size Register (DCU0_DISP_SIZE)	32	R/W	0000_0000h	55.5.7/3107
4005_801C	Horizontal Sync Parameter Register (DCU0_HSYN_PARA)	32	R/W	00C0_1803h	55.5.8/3108
4005_8020	Vertical Sync Parameter Register (DCU0_VSYN_PARA)	32	R/W	00C0_1803h	55.5.9/3109
4005_8024	Synchronize Polarity Register (DCU0_SYNPOL)	32	R/W	0000_0000h	55.5.10/ 3110
4005_8028	Threshold Register (DCU0_THRESHOLD)	32	R/W	0000_780Ah	55.5.11/ 3112
4005_802C	Interrupt Status Register (DCU0_INT_STATUS)	32	R/W	0000_0000h	55.5.12/ 3112
4005_8030	Interrupt Mask Register (DCU0_INT_MASK)	32	R/W	FCFF_5FFFh	55.5.13/ 3114
4005_8034	COLBAR_1 Register (DCU0_COLBAR_1)	32	R/W	FF00_0000h	55.5.14/ 3117
4005_8038	COLBAR_2 Register (DCU0_COLBAR_2)	32	R/W	FF00_00FFh	55.5.15/ 3118
4005_803C	COLBAR_3 Register (DCU0_COLBAR_3)	32	R/W	FF00_FFFFh	55.5.16/ 3118

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8040	COLBAR_4 Register (DCU0_COLBAR_4)	32	R/W	FF00_FF00h	55.5.17/ 3119
4005_8044	COLBAR_5 Register (DCU0_COLBAR_5)	32	R/W	FFFF_FF00h	55.5.18/ 3120
4005_8048	COLBAR_6 Register (DCU0_COLBAR_6)	32	R/W	FFFF_0000h	55.5.19/ 3120
4005_804C	COLBAR_7 Register (DCU0_COLBAR_7)	32	R/W	FFFF_00FFh	55.5.20/ 3121
4005_8050	COLBAR_8 Register (DCU0_COLBAR_8)	32	R/W	FFFF_FFFFh	55.5.21/ 3122
4005_8054	Divide Ratio Register (DCU0_DIV_RATIO)	32	R/W	0000_0000h	55.5.22/ 3122
4005_8058	Sign Calculation 1 Register (DCU0_SIGN_CALC_1)	32	R/W	0000_0000h	55.5.23/ 3123
4005_805C	Sign Calculation 2 Register (DCU0_SIGN_CALC_2)	32	R/W	0000_0000h	55.5.24/ 3123
4005_8060	CRC Value Register (DCU0_CRC_VAL)	32	R/W	0000_0000h	55.5.25/ 3124
4005_8064	PDI Status Register (DCU0_PDI_STATUS)	32	R/W	0000_0000h	55.5.26/ 3125
4005_8068	PDI Status Mask Register (DCU0_PDI_STA_MSK)	32	R/W	0000_03FFh	55.5.27/ 3126
4005_806C	Parameter Error Status 1 Register (DCU0_PARR_ERR_STATUS1)	32	R/W	0000_0000h	55.5.28/ 3127
4005_8070	Parameter Error Status 2 Register (DCU0_PARR_ERR_STATUS2)	32	R/W	0000_0000h	55.5.29/ 3128
4005_807C	Parameter Error Status 3 Register (DCU0_PARR_ERR_STATUS3)	32	R/W	0000_0000h	55.5.30/ 3129
4005_8080	Mask Parameter Error Status 1 Register (DCU0_MASK_PARR_ERR_STATUS1)	32	R/W	FFFF_FFFFh	55.5.31/ 3130
4005_8084	Mask Parameter Error Status 2 Register (DCU0_MASK_PARR_ERR_STATUS2)	32	R/W	FFFF_FFFFh	55.5.32/ 3130
4005_8090	Mask Parameter Error Status 3 Register (DCU0_MASK_PARR_ERR_STATUS3)	32	R/W	0007_FFFFh	55.5.33/ 3131
4005_8094	Threshold Input 1 Register (DCU0_THRESHOLD_INP_BUF_1)	32	R/W	7F00_7F00h	55.5.34/ 3132
4005_8098	Threshold Input 2 Register (DCU0_THRESHOLD_INP_BUF_2)	32	R/W	7F00_7F00h	55.5.35/ 3133
4005_809C	Threshold Input 3 Register (DCU0_THRESHOLD_INP_BUF_3)	32	R/W	7F00_7F00h	55.5.36/ 3134
4005_80A0	LUMA Component Register (DCU0_LUMA_COMP)	32	R/W	9512_A254h	55.5.37/ 3135
4005_80A4	Red Chroma Components Register (DCU0_CHROMA_RED)	32	R/W	0331_0000h	55.5.38/ 3136

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_80A8	Green Chroma Components Register (DCU0_CHROMA_GREEN)	32	R/W	0660_0F38h	55.5.39/ 3136
4005_80AC	Blue Chroma Components Register (DCU0_CHROMA_BLUE)	32	R/W	0000_0409h	55.5.40/ 3137
4005_80B0	CRC Position Register (DCU0_CRC_POS)	32	R/W	0000_0000h	55.5.41/ 3138
4005_80B4	Layer Interpolation Enable Register (DCU0_LYR_INTPOL_EN)	32	R/W	0000_0409h	55.5.42/ 3138
4005_80B8	Layer Luminance Component Register (DCU0_LYR_LUMA_COMP)	32	R/W	9512_A254h	55.5.43/ 3139
4005_80BC	Layer Chroma Red Register (DCU0_LYR_CHRM_RED)	32	R/W	0331_0000h	55.5.44/ 3140
4005_80C0	Layer Chroma Green Register (DCU0_LYR_CHRM_GRN)	32	R/W	0660_0F38h	55.5.45/ 3140
4005_80C4	Layer Chroma Blue Register (DCU0_LYR_CHRM_BLUE)	32	R/W	0000_0409h	55.5.46/ 3141
4005_80C8	Compression Image Size Register (DCU0_COMP_IMSIZE)	32	R/W	0000_0409h	55.5.47/ 3142
4005_80CC	Update Mode Register (DCU0_UPDATE_MODE)	32	R/W	0000_0000h	55.5.48/ 3142
4005_80D0	Underrun Register (DCU0_UNDERRUN)	32	R/W	0000_0000h	55.5.49/ 3143
4005_8100	Global Protection Register (DCU0_GLBL_PROTECT)	32	R/W	0000_0000h	55.5.50/ 3144
4005_8104	Soft Lock Bit Layer 0 Register (DCU0_SFT_LCK_BIT_L0)	32	R/W	0000_0000h	55.5.51/ 3145
4005_8108	Soft Lock Bit Layer 1 Register (DCU0_SFT_LCK_BIT_L1)	32	R/W	0000_0000h	55.5.52/ 3147
4005_810C	Soft Lock Display Size Register (DCU0_SFT_LCK_DISP_SIZE)	32	R/W	0000_0000h	55.5.53/ 3149
4005_8110	Soft Lock Hsync/Vsync Parameter Register (DCU0_SFT_LCK_HS_VS_PARA)	32	R/W	0000_0000h	55.5.54/ 3150
4005_8114	Soft Lock POL Register (DCU0_SFT_LCK_POL)	32	R/W	0000_0000h	55.5.55/ 3151
4005_8118	Soft Lock L0 Transparency Register (DCU0_SFT_LCK_L0_TRANSP)	32	R/W	0000_0000h	55.5.56/ 3152
4005_811C	Soft Lock L1 Transparency Register (DCU0_SFT_LCK_L1_TRANSP)	32	R/W	0000_0000h	55.5.57/ 3153
4005_8200	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL0_1)	32	R/W	See section	55.5.58/ 3154
4005_8204	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL0_2)	32	R/W	See section	55.5.59/ 3155
4005_8208	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL0_3)	32	R/W	See section	55.5.60/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_820C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL0_4)	32	R/W	See section	55.5.61/ 3156
4005_8210	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL0_5)	32	R/W	See section	55.5.62/ 3158
4005_8214	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL0_6)	32	R/W	See section	55.5.63/ 3159
4005_8218	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL0_7)	32	R/W	See section	55.5.64/ 3160
4005_821C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL0_8)	32	R/W	See section	55.5.65/ 3161
4005_8220	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL0_9)	32	R/W	See section	55.5.66/ 3161
4005_8240	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL1_1)	32	R/W	See section	55.5.58/ 3154
4005_8244	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL1_2)	32	R/W	See section	55.5.59/ 3155
4005_8248	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL1_3)	32	R/W	See section	55.5.60/ 3155
4005_824C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL1_4)	32	R/W	See section	55.5.61/ 3156
4005_8250	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL1_5)	32	R/W	See section	55.5.62/ 3158
4005_8254	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL1_6)	32	R/W	See section	55.5.63/ 3159
4005_8258	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL1_7)	32	R/W	See section	55.5.64/ 3160
4005_825C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL1_8)	32	R/W	See section	55.5.65/ 3161
4005_8260	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL1_9)	32	R/W	See section	55.5.66/ 3161
4005_8280	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL2_1)	32	R/W	See section	55.5.58/ 3154
4005_8284	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL2_2)	32	R/W	See section	55.5.59/ 3155
4005_8288	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL2_3)	32	R/W	See section	55.5.60/ 3155
4005_828C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL2_4)	32	R/W	See section	55.5.61/ 3156
4005_8290	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL2_5)	32	R/W	See section	55.5.62/ 3158
4005_8294	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL2_6)	32	R/W	See section	55.5.63/ 3159
4005_8298	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL2_7)	32	R/W	See section	55.5.64/ 3160

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_829C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL2_8)	32	R/W	See section	55.5.65/ 3161
4005_82A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL2_9)	32	R/W	See section	55.5.66/ 3161
4005_82C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL3_1)	32	R/W	See section	55.5.58/ 3154
4005_82C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL3_2)	32	R/W	See section	55.5.59/ 3155
4005_82C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL3_3)	32	R/W	See section	55.5.60/ 3155
4005_82CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL3_4)	32	R/W	See section	55.5.61/ 3156
4005_82D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL3_5)	32	R/W	See section	55.5.62/ 3158
4005_82D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL3_6)	32	R/W	See section	55.5.63/ 3159
4005_82D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL3_7)	32	R/W	See section	55.5.64/ 3160
4005_82DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL3_8)	32	R/W	See section	55.5.65/ 3161
4005_82E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL3_9)	32	R/W	See section	55.5.66/ 3161
4005_8300	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL4_1)	32	R/W	See section	55.5.58/ 3154
4005_8304	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL4_2)	32	R/W	See section	55.5.59/ 3155
4005_8308	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL4_3)	32	R/W	See section	55.5.60/ 3155
4005_830C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL4_4)	32	R/W	See section	55.5.61/ 3156
4005_8310	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL4_5)	32	R/W	See section	55.5.62/ 3158
4005_8314	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL4_6)	32	R/W	See section	55.5.63/ 3159
4005_8318	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL4_7)	32	R/W	See section	55.5.64/ 3160
4005_831C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL4_8)	32	R/W	See section	55.5.65/ 3161
4005_8320	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL4_9)	32	R/W	See section	55.5.66/ 3161
4005_8340	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL5_1)	32	R/W	See section	55.5.58/ 3154
4005_8344	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL5_2)	32	R/W	See section	55.5.59/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8348	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL5_3)	32	R/W	See section	55.5.60/ 3155
4005_834C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL5_4)	32	R/W	See section	55.5.61/ 3156
4005_8350	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL5_5)	32	R/W	See section	55.5.62/ 3158
4005_8354	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL5_6)	32	R/W	See section	55.5.63/ 3159
4005_8358	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL5_7)	32	R/W	See section	55.5.64/ 3160
4005_835C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL5_8)	32	R/W	See section	55.5.65/ 3161
4005_8360	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL5_9)	32	R/W	See section	55.5.66/ 3161
4005_8380	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL6_1)	32	R/W	See section	55.5.58/ 3154
4005_8384	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL6_2)	32	R/W	See section	55.5.59/ 3155
4005_8388	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL6_3)	32	R/W	See section	55.5.60/ 3155
4005_838C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL6_4)	32	R/W	See section	55.5.61/ 3156
4005_8390	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL6_5)	32	R/W	See section	55.5.62/ 3158
4005_8394	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL6_6)	32	R/W	See section	55.5.63/ 3159
4005_8398	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL6_7)	32	R/W	See section	55.5.64/ 3160
4005_839C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL6_8)	32	R/W	See section	55.5.65/ 3161
4005_83A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL6_9)	32	R/W	See section	55.5.66/ 3161
4005_83C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL7_1)	32	R/W	See section	55.5.58/ 3154
4005_83C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL7_2)	32	R/W	See section	55.5.59/ 3155
4005_83C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL7_3)	32	R/W	See section	55.5.60/ 3155
4005_83CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL7_4)	32	R/W	See section	55.5.61/ 3156
4005_83D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL7_5)	32	R/W	See section	55.5.62/ 3158
4005_83D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL7_6)	32	R/W	See section	55.5.63/ 3159

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_83D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL7_7)	32	R/W	See section	55.5.64/ 3160
4005_83DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL7_8)	32	R/W	See section	55.5.65/ 3161
4005_83E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL7_9)	32	R/W	See section	55.5.66/ 3161
4005_8400	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL8_1)	32	R/W	See section	55.5.58/ 3154
4005_8404	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL8_2)	32	R/W	See section	55.5.59/ 3155
4005_8408	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL8_3)	32	R/W	See section	55.5.60/ 3155
4005_840C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL8_4)	32	R/W	See section	55.5.61/ 3156
4005_8410	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL8_5)	32	R/W	See section	55.5.62/ 3158
4005_8414	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL8_6)	32	R/W	See section	55.5.63/ 3159
4005_8418	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL8_7)	32	R/W	See section	55.5.64/ 3160
4005_841C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL8_8)	32	R/W	See section	55.5.65/ 3161
4005_8420	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL8_9)	32	R/W	See section	55.5.66/ 3161
4005_8440	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL9_1)	32	R/W	See section	55.5.58/ 3154
4005_8444	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL9_2)	32	R/W	See section	55.5.59/ 3155
4005_8448	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL9_3)	32	R/W	See section	55.5.60/ 3155
4005_844C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL9_4)	32	R/W	See section	55.5.61/ 3156
4005_8450	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL9_5)	32	R/W	See section	55.5.62/ 3158
4005_8454	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL9_6)	32	R/W	See section	55.5.63/ 3159
4005_8458	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL9_7)	32	R/W	See section	55.5.64/ 3160
4005_845C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL9_8)	32	R/W	See section	55.5.65/ 3161
4005_8460	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL9_9)	32	R/W	See section	55.5.66/ 3161
4005_8480	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL10_1)	32	R/W	See section	55.5.58/ 3154

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8484	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL10_2)	32	R/W	See section	55.5.59/ 3155
4005_8488	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL10_3)	32	R/W	See section	55.5.60/ 3155
4005_848C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL10_4)	32	R/W	See section	55.5.61/ 3156
4005_8490	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL10_5)	32	R/W	See section	55.5.62/ 3158
4005_8494	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL10_6)	32	R/W	See section	55.5.63/ 3159
4005_8498	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL10_7)	32	R/W	See section	55.5.64/ 3160
4005_849C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL10_8)	32	R/W	See section	55.5.65/ 3161
4005_84A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL10_9)	32	R/W	See section	55.5.66/ 3161
4005_84C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL11_1)	32	R/W	See section	55.5.58/ 3154
4005_84C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL11_2)	32	R/W	See section	55.5.59/ 3155
4005_84C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL11_3)	32	R/W	See section	55.5.60/ 3155
4005_84CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL11_4)	32	R/W	See section	55.5.61/ 3156
4005_84D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL11_5)	32	R/W	See section	55.5.62/ 3158
4005_84D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL11_6)	32	R/W	See section	55.5.63/ 3159
4005_84D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL11_7)	32	R/W	See section	55.5.64/ 3160
4005_84DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL11_8)	32	R/W	See section	55.5.65/ 3161
4005_84E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL11_9)	32	R/W	See section	55.5.66/ 3161
4005_8500	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL12_1)	32	R/W	See section	55.5.58/ 3154
4005_8504	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL12_2)	32	R/W	See section	55.5.59/ 3155
4005_8508	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL12_3)	32	R/W	See section	55.5.60/ 3155
4005_850C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL12_4)	32	R/W	See section	55.5.61/ 3156
4005_8510	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL12_5)	32	R/W	See section	55.5.62/ 3158

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8514	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL12_6)	32	R/W	See section	55.5.63/ 3159
4005_8518	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL12_7)	32	R/W	See section	55.5.64/ 3160
4005_851C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL12_8)	32	R/W	See section	55.5.65/ 3161
4005_8520	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL12_9)	32	R/W	See section	55.5.66/ 3161
4005_8540	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL13_1)	32	R/W	See section	55.5.58/ 3154
4005_8544	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL13_2)	32	R/W	See section	55.5.59/ 3155
4005_8548	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL13_3)	32	R/W	See section	55.5.60/ 3155
4005_854C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL13_4)	32	R/W	See section	55.5.61/ 3156
4005_8550	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL13_5)	32	R/W	See section	55.5.62/ 3158
4005_8554	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL13_6)	32	R/W	See section	55.5.63/ 3159
4005_8558	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL13_7)	32	R/W	See section	55.5.64/ 3160
4005_855C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL13_8)	32	R/W	See section	55.5.65/ 3161
4005_8560	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL13_9)	32	R/W	See section	55.5.66/ 3161
4005_8580	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL14_1)	32	R/W	See section	55.5.58/ 3154
4005_8584	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL14_2)	32	R/W	See section	55.5.59/ 3155
4005_8588	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL14_3)	32	R/W	See section	55.5.60/ 3155
4005_858C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL14_4)	32	R/W	See section	55.5.61/ 3156
4005_8590	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL14_5)	32	R/W	See section	55.5.62/ 3158
4005_8594	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL14_6)	32	R/W	See section	55.5.63/ 3159
4005_8598	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL14_7)	32	R/W	See section	55.5.64/ 3160
4005_859C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL14_8)	32	R/W	See section	55.5.65/ 3161
4005_85A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL14_9)	32	R/W	See section	55.5.66/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_85C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL15_1)	32	R/W	See section	55.5.58/ 3154
4005_85C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL15_2)	32	R/W	See section	55.5.59/ 3155
4005_85C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL15_3)	32	R/W	See section	55.5.60/ 3155
4005_85CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL15_4)	32	R/W	See section	55.5.61/ 3156
4005_85D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL15_5)	32	R/W	See section	55.5.62/ 3158
4005_85D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL15_6)	32	R/W	See section	55.5.63/ 3159
4005_85D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL15_7)	32	R/W	See section	55.5.64/ 3160
4005_85DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL15_8)	32	R/W	See section	55.5.65/ 3161
4005_85E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL15_9)	32	R/W	See section	55.5.66/ 3161
4005_8600	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL16_1)	32	R/W	See section	55.5.58/ 3154
4005_8604	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL16_2)	32	R/W	See section	55.5.59/ 3155
4005_8608	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL16_3)	32	R/W	See section	55.5.60/ 3155
4005_860C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL16_4)	32	R/W	See section	55.5.61/ 3156
4005_8610	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL16_5)	32	R/W	See section	55.5.62/ 3158
4005_8614	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL16_6)	32	R/W	See section	55.5.63/ 3159
4005_8618	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL16_7)	32	R/W	See section	55.5.64/ 3160
4005_861C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL16_8)	32	R/W	See section	55.5.65/ 3161
4005_8620	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL16_9)	32	R/W	See section	55.5.66/ 3161
4005_8640	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL17_1)	32	R/W	See section	55.5.58/ 3154
4005_8644	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL17_2)	32	R/W	See section	55.5.59/ 3155
4005_8648	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL17_3)	32	R/W	See section	55.5.60/ 3155
4005_864C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL17_4)	32	R/W	See section	55.5.61/ 3156

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8650	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL17_5)	32	R/W	See section	55.5.62/ 3158
4005_8654	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL17_6)	32	R/W	See section	55.5.63/ 3159
4005_8658	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL17_7)	32	R/W	See section	55.5.64/ 3160
4005_865C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL17_8)	32	R/W	See section	55.5.65/ 3161
4005_8660	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL17_9)	32	R/W	See section	55.5.66/ 3161
4005_8680	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL18_1)	32	R/W	See section	55.5.58/ 3154
4005_8684	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL18_2)	32	R/W	See section	55.5.59/ 3155
4005_8688	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL18_3)	32	R/W	See section	55.5.60/ 3155
4005_868C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL18_4)	32	R/W	See section	55.5.61/ 3156
4005_8690	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL18_5)	32	R/W	See section	55.5.62/ 3158
4005_8694	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL18_6)	32	R/W	See section	55.5.63/ 3159
4005_8698	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL18_7)	32	R/W	See section	55.5.64/ 3160
4005_869C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL18_8)	32	R/W	See section	55.5.65/ 3161
4005_86A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL18_9)	32	R/W	See section	55.5.66/ 3161
4005_86C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL19_1)	32	R/W	See section	55.5.58/ 3154
4005_86C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL19_2)	32	R/W	See section	55.5.59/ 3155
4005_86C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL19_3)	32	R/W	See section	55.5.60/ 3155
4005_86CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL19_4)	32	R/W	See section	55.5.61/ 3156
4005_86D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL19_5)	32	R/W	See section	55.5.62/ 3158
4005_86D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL19_6)	32	R/W	See section	55.5.63/ 3159
4005_86D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL19_7)	32	R/W	See section	55.5.64/ 3160
4005_86DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL19_8)	32	R/W	See section	55.5.65/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_86E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL19_9)	32	R/W	See section	55.5.66/ 3161
4005_8700	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL20_1)	32	R/W	See section	55.5.58/ 3154
4005_8704	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL20_2)	32	R/W	See section	55.5.59/ 3155
4005_8708	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL20_3)	32	R/W	See section	55.5.60/ 3155
4005_870C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL20_4)	32	R/W	See section	55.5.61/ 3156
4005_8710	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL20_5)	32	R/W	See section	55.5.62/ 3158
4005_8714	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL20_6)	32	R/W	See section	55.5.63/ 3159
4005_8718	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL20_7)	32	R/W	See section	55.5.64/ 3160
4005_871C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL20_8)	32	R/W	See section	55.5.65/ 3161
4005_8720	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL20_9)	32	R/W	See section	55.5.66/ 3161
4005_8740	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL21_1)	32	R/W	See section	55.5.58/ 3154
4005_8744	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL21_2)	32	R/W	See section	55.5.59/ 3155
4005_8748	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL21_3)	32	R/W	See section	55.5.60/ 3155
4005_874C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL21_4)	32	R/W	See section	55.5.61/ 3156
4005_8750	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL21_5)	32	R/W	See section	55.5.62/ 3158
4005_8754	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL21_6)	32	R/W	See section	55.5.63/ 3159
4005_8758	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL21_7)	32	R/W	See section	55.5.64/ 3160
4005_875C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL21_8)	32	R/W	See section	55.5.65/ 3161
4005_8760	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL21_9)	32	R/W	See section	55.5.66/ 3161
4005_8780	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL22_1)	32	R/W	See section	55.5.58/ 3154
4005_8784	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL22_2)	32	R/W	See section	55.5.59/ 3155
4005_8788	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL22_3)	32	R/W	See section	55.5.60/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_878C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL22_4)	32	R/W	See section	55.5.61/ 3156
4005_8790	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL22_5)	32	R/W	See section	55.5.62/ 3158
4005_8794	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL22_6)	32	R/W	See section	55.5.63/ 3159
4005_8798	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL22_7)	32	R/W	See section	55.5.64/ 3160
4005_879C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL22_8)	32	R/W	See section	55.5.65/ 3161
4005_87A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL22_9)	32	R/W	See section	55.5.66/ 3161
4005_87C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL23_1)	32	R/W	See section	55.5.58/ 3154
4005_87C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL23_2)	32	R/W	See section	55.5.59/ 3155
4005_87C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL23_3)	32	R/W	See section	55.5.60/ 3155
4005_87CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL23_4)	32	R/W	See section	55.5.61/ 3156
4005_87D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL23_5)	32	R/W	See section	55.5.62/ 3158
4005_87D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL23_6)	32	R/W	See section	55.5.63/ 3159
4005_87D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL23_7)	32	R/W	See section	55.5.64/ 3160
4005_87DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL23_8)	32	R/W	See section	55.5.65/ 3161
4005_87E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL23_9)	32	R/W	See section	55.5.66/ 3161
4005_8800	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL24_1)	32	R/W	See section	55.5.58/ 3154
4005_8804	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL24_2)	32	R/W	See section	55.5.59/ 3155
4005_8808	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL24_3)	32	R/W	See section	55.5.60/ 3155
4005_880C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL24_4)	32	R/W	See section	55.5.61/ 3156
4005_8810	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL24_5)	32	R/W	See section	55.5.62/ 3158
4005_8814	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL24_6)	32	R/W	See section	55.5.63/ 3159
4005_8818	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL24_7)	32	R/W	See section	55.5.64/ 3160

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_881C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL24_8)	32	R/W	See section	55.5.65/ 3161
4005_8820	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL24_9)	32	R/W	See section	55.5.66/ 3161
4005_8840	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL25_1)	32	R/W	See section	55.5.58/ 3154
4005_8844	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL25_2)	32	R/W	See section	55.5.59/ 3155
4005_8848	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL25_3)	32	R/W	See section	55.5.60/ 3155
4005_884C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL25_4)	32	R/W	See section	55.5.61/ 3156
4005_8850	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL25_5)	32	R/W	See section	55.5.62/ 3158
4005_8854	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL25_6)	32	R/W	See section	55.5.63/ 3159
4005_8858	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL25_7)	32	R/W	See section	55.5.64/ 3160
4005_885C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL25_8)	32	R/W	See section	55.5.65/ 3161
4005_8860	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL25_9)	32	R/W	See section	55.5.66/ 3161
4005_8880	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL26_1)	32	R/W	See section	55.5.58/ 3154
4005_8884	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL26_2)	32	R/W	See section	55.5.59/ 3155
4005_8888	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL26_3)	32	R/W	See section	55.5.60/ 3155
4005_888C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL26_4)	32	R/W	See section	55.5.61/ 3156
4005_8890	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL26_5)	32	R/W	See section	55.5.62/ 3158
4005_8894	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL26_6)	32	R/W	See section	55.5.63/ 3159
4005_8898	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL26_7)	32	R/W	See section	55.5.64/ 3160
4005_889C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL26_8)	32	R/W	See section	55.5.65/ 3161
4005_88A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL26_9)	32	R/W	See section	55.5.66/ 3161
4005_88C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL27_1)	32	R/W	See section	55.5.58/ 3154
4005_88C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL27_2)	32	R/W	See section	55.5.59/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_88C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL27_3)	32	R/W	See section	55.5.60/ 3155
4005_88CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL27_4)	32	R/W	See section	55.5.61/ 3156
4005_88D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL27_5)	32	R/W	See section	55.5.62/ 3158
4005_88D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL27_6)	32	R/W	See section	55.5.63/ 3159
4005_88D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL27_7)	32	R/W	See section	55.5.64/ 3160
4005_88DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL27_8)	32	R/W	See section	55.5.65/ 3161
4005_88E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL27_9)	32	R/W	See section	55.5.66/ 3161
4005_8900	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL28_1)	32	R/W	See section	55.5.58/ 3154
4005_8904	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL28_2)	32	R/W	See section	55.5.59/ 3155
4005_8908	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL28_3)	32	R/W	See section	55.5.60/ 3155
4005_890C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL28_4)	32	R/W	See section	55.5.61/ 3156
4005_8910	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL28_5)	32	R/W	See section	55.5.62/ 3158
4005_8914	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL28_6)	32	R/W	See section	55.5.63/ 3159
4005_8918	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL28_7)	32	R/W	See section	55.5.64/ 3160
4005_891C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL28_8)	32	R/W	See section	55.5.65/ 3161
4005_8920	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL28_9)	32	R/W	See section	55.5.66/ 3161
4005_8940	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL29_1)	32	R/W	See section	55.5.58/ 3154
4005_8944	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL29_2)	32	R/W	See section	55.5.59/ 3155
4005_8948	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL29_3)	32	R/W	See section	55.5.60/ 3155
4005_894C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL29_4)	32	R/W	See section	55.5.61/ 3156
4005_8950	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL29_5)	32	R/W	See section	55.5.62/ 3158
4005_8954	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL29_6)	32	R/W	See section	55.5.63/ 3159

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8958	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL29_7)	32	R/W	See section	55.5.64/ 3160
4005_895C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL29_8)	32	R/W	See section	55.5.65/ 3161
4005_8960	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL29_9)	32	R/W	See section	55.5.66/ 3161
4005_8980	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL30_1)	32	R/W	See section	55.5.58/ 3154
4005_8984	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL30_2)	32	R/W	See section	55.5.59/ 3155
4005_8988	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL30_3)	32	R/W	See section	55.5.60/ 3155
4005_898C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL30_4)	32	R/W	See section	55.5.61/ 3156
4005_8990	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL30_5)	32	R/W	See section	55.5.62/ 3158
4005_8994	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL30_6)	32	R/W	See section	55.5.63/ 3159
4005_8998	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL30_7)	32	R/W	See section	55.5.64/ 3160
4005_899C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL30_8)	32	R/W	See section	55.5.65/ 3161
4005_89A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL30_9)	32	R/W	See section	55.5.66/ 3161
4005_89C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL31_1)	32	R/W	See section	55.5.58/ 3154
4005_89C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL31_2)	32	R/W	See section	55.5.59/ 3155
4005_89C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL31_3)	32	R/W	See section	55.5.60/ 3155
4005_89CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL31_4)	32	R/W	See section	55.5.61/ 3156
4005_89D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL31_5)	32	R/W	See section	55.5.62/ 3158
4005_89D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL31_6)	32	R/W	See section	55.5.63/ 3159
4005_89D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL31_7)	32	R/W	See section	55.5.64/ 3160
4005_89DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL31_8)	32	R/W	See section	55.5.65/ 3161
4005_89E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL31_9)	32	R/W	See section	55.5.66/ 3161
4005_8A00	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL32_1)	32	R/W	See section	55.5.58/ 3154

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8A04	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL32_2)	32	R/W	See section	55.5.59/ 3155
4005_8A08	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL32_3)	32	R/W	See section	55.5.60/ 3155
4005_8A0C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL32_4)	32	R/W	See section	55.5.61/ 3156
4005_8A10	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL32_5)	32	R/W	See section	55.5.62/ 3158
4005_8A14	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL32_6)	32	R/W	See section	55.5.63/ 3159
4005_8A18	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL32_7)	32	R/W	See section	55.5.64/ 3160
4005_8A1C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL32_8)	32	R/W	See section	55.5.65/ 3161
4005_8A20	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL32_9)	32	R/W	See section	55.5.66/ 3161
4005_8A40	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL33_1)	32	R/W	See section	55.5.58/ 3154
4005_8A44	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL33_2)	32	R/W	See section	55.5.59/ 3155
4005_8A48	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL33_3)	32	R/W	See section	55.5.60/ 3155
4005_8A4C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL33_4)	32	R/W	See section	55.5.61/ 3156
4005_8A50	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL33_5)	32	R/W	See section	55.5.62/ 3158
4005_8A54	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL33_6)	32	R/W	See section	55.5.63/ 3159
4005_8A58	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL33_7)	32	R/W	See section	55.5.64/ 3160
4005_8A5C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL33_8)	32	R/W	See section	55.5.65/ 3161
4005_8A60	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL33_9)	32	R/W	See section	55.5.66/ 3161
4005_8A80	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL34_1)	32	R/W	See section	55.5.58/ 3154
4005_8A84	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL34_2)	32	R/W	See section	55.5.59/ 3155
4005_8A88	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL34_3)	32	R/W	See section	55.5.60/ 3155
4005_8A8C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL34_4)	32	R/W	See section	55.5.61/ 3156
4005_8A90	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL34_5)	32	R/W	See section	55.5.62/ 3158

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8A94	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL34_6)	32	R/W	See section	55.5.63/ 3159
4005_8A98	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL34_7)	32	R/W	See section	55.5.64/ 3160
4005_8A9C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL34_8)	32	R/W	See section	55.5.65/ 3161
4005_8AA0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL34_9)	32	R/W	See section	55.5.66/ 3161
4005_8AC0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL35_1)	32	R/W	See section	55.5.58/ 3154
4005_8AC4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL35_2)	32	R/W	See section	55.5.59/ 3155
4005_8AC8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL35_3)	32	R/W	See section	55.5.60/ 3155
4005_8ACC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL35_4)	32	R/W	See section	55.5.61/ 3156
4005_8AD0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL35_5)	32	R/W	See section	55.5.62/ 3158
4005_8AD4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL35_6)	32	R/W	See section	55.5.63/ 3159
4005_8AD8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL35_7)	32	R/W	See section	55.5.64/ 3160
4005_8ADC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL35_8)	32	R/W	See section	55.5.65/ 3161
4005_8AE0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL35_9)	32	R/W	See section	55.5.66/ 3161
4005_8B00	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL36_1)	32	R/W	See section	55.5.58/ 3154
4005_8B04	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL36_2)	32	R/W	See section	55.5.59/ 3155
4005_8B08	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL36_3)	32	R/W	See section	55.5.60/ 3155
4005_8B0C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL36_4)	32	R/W	See section	55.5.61/ 3156
4005_8B10	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL36_5)	32	R/W	See section	55.5.62/ 3158
4005_8B14	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL36_6)	32	R/W	See section	55.5.63/ 3159
4005_8B18	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL36_7)	32	R/W	See section	55.5.64/ 3160
4005_8B1C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL36_8)	32	R/W	See section	55.5.65/ 3161
4005_8B20	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL36_9)	32	R/W	See section	55.5.66/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8B40	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL37_1)	32	R/W	See section	55.5.58/ 3154
4005_8B44	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL37_2)	32	R/W	See section	55.5.59/ 3155
4005_8B48	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL37_3)	32	R/W	See section	55.5.60/ 3155
4005_8B4C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL37_4)	32	R/W	See section	55.5.61/ 3156
4005_8B50	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL37_5)	32	R/W	See section	55.5.62/ 3158
4005_8B54	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL37_6)	32	R/W	See section	55.5.63/ 3159
4005_8B58	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL37_7)	32	R/W	See section	55.5.64/ 3160
4005_8B5C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL37_8)	32	R/W	See section	55.5.65/ 3161
4005_8B60	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL37_9)	32	R/W	See section	55.5.66/ 3161
4005_8B80	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL38_1)	32	R/W	See section	55.5.58/ 3154
4005_8B84	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL38_2)	32	R/W	See section	55.5.59/ 3155
4005_8B88	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL38_3)	32	R/W	See section	55.5.60/ 3155
4005_8B8C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL38_4)	32	R/W	See section	55.5.61/ 3156
4005_8B90	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL38_5)	32	R/W	See section	55.5.62/ 3158
4005_8B94	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL38_6)	32	R/W	See section	55.5.63/ 3159
4005_8B98	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL38_7)	32	R/W	See section	55.5.64/ 3160
4005_8B9C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL38_8)	32	R/W	See section	55.5.65/ 3161
4005_8BA0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL38_9)	32	R/W	See section	55.5.66/ 3161
4005_8BC0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL39_1)	32	R/W	See section	55.5.58/ 3154
4005_8BC4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL39_2)	32	R/W	See section	55.5.59/ 3155
4005_8BC8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL39_3)	32	R/W	See section	55.5.60/ 3155
4005_8BCC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL39_4)	32	R/W	See section	55.5.61/ 3156

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8BD0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL39_5)	32	R/W	See section	55.5.62/ 3158
4005_8BD4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL39_6)	32	R/W	See section	55.5.63/ 3159
4005_8BD8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL39_7)	32	R/W	See section	55.5.64/ 3160
4005_8BDC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL39_8)	32	R/W	See section	55.5.65/ 3161
4005_8BE0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL39_9)	32	R/W	See section	55.5.66/ 3161
4005_8C00	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL40_1)	32	R/W	See section	55.5.58/ 3154
4005_8C04	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL40_2)	32	R/W	See section	55.5.59/ 3155
4005_8C08	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL40_3)	32	R/W	See section	55.5.60/ 3155
4005_8C0C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL40_4)	32	R/W	See section	55.5.61/ 3156
4005_8C10	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL40_5)	32	R/W	See section	55.5.62/ 3158
4005_8C14	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL40_6)	32	R/W	See section	55.5.63/ 3159
4005_8C18	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL40_7)	32	R/W	See section	55.5.64/ 3160
4005_8C1C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL40_8)	32	R/W	See section	55.5.65/ 3161
4005_8C20	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL40_9)	32	R/W	See section	55.5.66/ 3161
4005_8C40	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL41_1)	32	R/W	See section	55.5.58/ 3154
4005_8C44	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL41_2)	32	R/W	See section	55.5.59/ 3155
4005_8C48	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL41_3)	32	R/W	See section	55.5.60/ 3155
4005_8C4C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL41_4)	32	R/W	See section	55.5.61/ 3156
4005_8C50	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL41_5)	32	R/W	See section	55.5.62/ 3158
4005_8C54	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL41_6)	32	R/W	See section	55.5.63/ 3159
4005_8C58	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL41_7)	32	R/W	See section	55.5.64/ 3160
4005_8C5C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL41_8)	32	R/W	See section	55.5.65/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8C60	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL41_9)	32	R/W	See section	55.5.66/ 3161
4005_8C80	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL42_1)	32	R/W	See section	55.5.58/ 3154
4005_8C84	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL42_2)	32	R/W	See section	55.5.59/ 3155
4005_8C88	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL42_3)	32	R/W	See section	55.5.60/ 3155
4005_8C8C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL42_4)	32	R/W	See section	55.5.61/ 3156
4005_8C90	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL42_5)	32	R/W	See section	55.5.62/ 3158
4005_8C94	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL42_6)	32	R/W	See section	55.5.63/ 3159
4005_8C98	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL42_7)	32	R/W	See section	55.5.64/ 3160
4005_8C9C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL42_8)	32	R/W	See section	55.5.65/ 3161
4005_8CA0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL42_9)	32	R/W	See section	55.5.66/ 3161
4005_8CC0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL43_1)	32	R/W	See section	55.5.58/ 3154
4005_8CC4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL43_2)	32	R/W	See section	55.5.59/ 3155
4005_8CC8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL43_3)	32	R/W	See section	55.5.60/ 3155
4005_8CCC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL43_4)	32	R/W	See section	55.5.61/ 3156
4005_8CD0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL43_5)	32	R/W	See section	55.5.62/ 3158
4005_8CD4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL43_6)	32	R/W	See section	55.5.63/ 3159
4005_8CD8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL43_7)	32	R/W	See section	55.5.64/ 3160
4005_8CDC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL43_8)	32	R/W	See section	55.5.65/ 3161
4005_8CE0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL43_9)	32	R/W	See section	55.5.66/ 3161
4005_8D00	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL44_1)	32	R/W	See section	55.5.58/ 3154
4005_8D04	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL44_2)	32	R/W	See section	55.5.59/ 3155
4005_8D08	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL44_3)	32	R/W	See section	55.5.60/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8D0C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL44_4)	32	R/W	See section	55.5.61/ 3156
4005_8D10	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL44_5)	32	R/W	See section	55.5.62/ 3158
4005_8D14	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL44_6)	32	R/W	See section	55.5.63/ 3159
4005_8D18	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL44_7)	32	R/W	See section	55.5.64/ 3160
4005_8D1C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL44_8)	32	R/W	See section	55.5.65/ 3161
4005_8D20	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL44_9)	32	R/W	See section	55.5.66/ 3161
4005_8D40	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL45_1)	32	R/W	See section	55.5.58/ 3154
4005_8D44	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL45_2)	32	R/W	See section	55.5.59/ 3155
4005_8D48	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL45_3)	32	R/W	See section	55.5.60/ 3155
4005_8D4C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL45_4)	32	R/W	See section	55.5.61/ 3156
4005_8D50	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL45_5)	32	R/W	See section	55.5.62/ 3158
4005_8D54	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL45_6)	32	R/W	See section	55.5.63/ 3159
4005_8D58	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL45_7)	32	R/W	See section	55.5.64/ 3160
4005_8D5C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL45_8)	32	R/W	See section	55.5.65/ 3161
4005_8D60	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL45_9)	32	R/W	See section	55.5.66/ 3161
4005_8D80	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL46_1)	32	R/W	See section	55.5.58/ 3154
4005_8D84	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL46_2)	32	R/W	See section	55.5.59/ 3155
4005_8D88	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL46_3)	32	R/W	See section	55.5.60/ 3155
4005_8D8C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL46_4)	32	R/W	See section	55.5.61/ 3156
4005_8D90	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL46_5)	32	R/W	See section	55.5.62/ 3158
4005_8D94	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL46_6)	32	R/W	See section	55.5.63/ 3159
4005_8D98	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL46_7)	32	R/W	See section	55.5.64/ 3160

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8D9C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL46_8)	32	R/W	See section	55.5.65/ 3161
4005_8DA0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL46_9)	32	R/W	See section	55.5.66/ 3161
4005_8DC0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL47_1)	32	R/W	See section	55.5.58/ 3154
4005_8DC4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL47_2)	32	R/W	See section	55.5.59/ 3155
4005_8DC8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL47_3)	32	R/W	See section	55.5.60/ 3155
4005_8DCC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL47_4)	32	R/W	See section	55.5.61/ 3156
4005_8DD0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL47_5)	32	R/W	See section	55.5.62/ 3158
4005_8DD4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL47_6)	32	R/W	See section	55.5.63/ 3159
4005_8DD8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL47_7)	32	R/W	See section	55.5.64/ 3160
4005_8DDC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL47_8)	32	R/W	See section	55.5.65/ 3161
4005_8DE0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL47_9)	32	R/W	See section	55.5.66/ 3161
4005_8E00	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL48_1)	32	R/W	See section	55.5.58/ 3154
4005_8E04	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL48_2)	32	R/W	See section	55.5.59/ 3155
4005_8E08	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL48_3)	32	R/W	See section	55.5.60/ 3155
4005_8E0C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL48_4)	32	R/W	See section	55.5.61/ 3156
4005_8E10	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL48_5)	32	R/W	See section	55.5.62/ 3158
4005_8E14	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL48_6)	32	R/W	See section	55.5.63/ 3159
4005_8E18	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL48_7)	32	R/W	See section	55.5.64/ 3160
4005_8E1C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL48_8)	32	R/W	See section	55.5.65/ 3161
4005_8E20	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL48_9)	32	R/W	See section	55.5.66/ 3161
4005_8E40	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL49_1)	32	R/W	See section	55.5.58/ 3154
4005_8E44	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL49_2)	32	R/W	See section	55.5.59/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8E48	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL49_3)	32	R/W	See section	55.5.60/ 3155
4005_8E4C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL49_4)	32	R/W	See section	55.5.61/ 3156
4005_8E50	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL49_5)	32	R/W	See section	55.5.62/ 3158
4005_8E54	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL49_6)	32	R/W	See section	55.5.63/ 3159
4005_8E58	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL49_7)	32	R/W	See section	55.5.64/ 3160
4005_8E5C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL49_8)	32	R/W	See section	55.5.65/ 3161
4005_8E60	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL49_9)	32	R/W	See section	55.5.66/ 3161
4005_8E80	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL50_1)	32	R/W	See section	55.5.58/ 3154
4005_8E84	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL50_2)	32	R/W	See section	55.5.59/ 3155
4005_8E88	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL50_3)	32	R/W	See section	55.5.60/ 3155
4005_8E8C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL50_4)	32	R/W	See section	55.5.61/ 3156
4005_8E90	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL50_5)	32	R/W	See section	55.5.62/ 3158
4005_8E94	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL50_6)	32	R/W	See section	55.5.63/ 3159
4005_8E98	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL50_7)	32	R/W	See section	55.5.64/ 3160
4005_8E9C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL50_8)	32	R/W	See section	55.5.65/ 3161
4005_8EA0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL50_9)	32	R/W	See section	55.5.66/ 3161
4005_8EC0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL51_1)	32	R/W	See section	55.5.58/ 3154
4005_8EC4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL51_2)	32	R/W	See section	55.5.59/ 3155
4005_8EC8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL51_3)	32	R/W	See section	55.5.60/ 3155
4005_8ECC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL51_4)	32	R/W	See section	55.5.61/ 3156
4005_8ED0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL51_5)	32	R/W	See section	55.5.62/ 3158
4005_8ED4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL51_6)	32	R/W	See section	55.5.63/ 3159

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8ED8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL51_7)	32	R/W	See section	55.5.64/ 3160
4005_8EDC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL51_8)	32	R/W	See section	55.5.65/ 3161
4005_8EE0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL51_9)	32	R/W	See section	55.5.66/ 3161
4005_8F00	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL52_1)	32	R/W	See section	55.5.58/ 3154
4005_8F04	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL52_2)	32	R/W	See section	55.5.59/ 3155
4005_8F08	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL52_3)	32	R/W	See section	55.5.60/ 3155
4005_8F0C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL52_4)	32	R/W	See section	55.5.61/ 3156
4005_8F10	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL52_5)	32	R/W	See section	55.5.62/ 3158
4005_8F14	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL52_6)	32	R/W	See section	55.5.63/ 3159
4005_8F18	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL52_7)	32	R/W	See section	55.5.64/ 3160
4005_8F1C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL52_8)	32	R/W	See section	55.5.65/ 3161
4005_8F20	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL52_9)	32	R/W	See section	55.5.66/ 3161
4005_8F40	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL53_1)	32	R/W	See section	55.5.58/ 3154
4005_8F44	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL53_2)	32	R/W	See section	55.5.59/ 3155
4005_8F48	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL53_3)	32	R/W	See section	55.5.60/ 3155
4005_8F4C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL53_4)	32	R/W	See section	55.5.61/ 3156
4005_8F50	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL53_5)	32	R/W	See section	55.5.62/ 3158
4005_8F54	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL53_6)	32	R/W	See section	55.5.63/ 3159
4005_8F58	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL53_7)	32	R/W	See section	55.5.64/ 3160
4005_8F5C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL53_8)	32	R/W	See section	55.5.65/ 3161
4005_8F60	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL53_9)	32	R/W	See section	55.5.66/ 3161
4005_8F80	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL54_1)	32	R/W	See section	55.5.58/ 3154

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_8F84	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL54_2)	32	R/W	See section	55.5.59/ 3155
4005_8F88	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL54_3)	32	R/W	See section	55.5.60/ 3155
4005_8F8C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL54_4)	32	R/W	See section	55.5.61/ 3156
4005_8F90	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL54_5)	32	R/W	See section	55.5.62/ 3158
4005_8F94	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL54_6)	32	R/W	See section	55.5.63/ 3159
4005_8F98	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL54_7)	32	R/W	See section	55.5.64/ 3160
4005_8F9C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL54_8)	32	R/W	See section	55.5.65/ 3161
4005_8FA0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL54_9)	32	R/W	See section	55.5.66/ 3161
4005_8FC0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL55_1)	32	R/W	See section	55.5.58/ 3154
4005_8FC4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL55_2)	32	R/W	See section	55.5.59/ 3155
4005_8FC8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL55_3)	32	R/W	See section	55.5.60/ 3155
4005_8FCC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL55_4)	32	R/W	See section	55.5.61/ 3156
4005_8FD0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL55_5)	32	R/W	See section	55.5.62/ 3158
4005_8FD4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL55_6)	32	R/W	See section	55.5.63/ 3159
4005_8FD8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL55_7)	32	R/W	See section	55.5.64/ 3160
4005_8FDC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL55_8)	32	R/W	See section	55.5.65/ 3161
4005_8FE0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL55_9)	32	R/W	See section	55.5.66/ 3161
4005_9000	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL56_1)	32	R/W	See section	55.5.58/ 3154
4005_9004	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL56_2)	32	R/W	See section	55.5.59/ 3155
4005_9008	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL56_3)	32	R/W	See section	55.5.60/ 3155
4005_900C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL56_4)	32	R/W	See section	55.5.61/ 3156
4005_9010	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL56_5)	32	R/W	See section	55.5.62/ 3158

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_9014	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL56_6)	32	R/W	See section	55.5.63/ 3159
4005_9018	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL56_7)	32	R/W	See section	55.5.64/ 3160
4005_901C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL56_8)	32	R/W	See section	55.5.65/ 3161
4005_9020	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL56_9)	32	R/W	See section	55.5.66/ 3161
4005_9040	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL57_1)	32	R/W	See section	55.5.58/ 3154
4005_9044	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL57_2)	32	R/W	See section	55.5.59/ 3155
4005_9048	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL57_3)	32	R/W	See section	55.5.60/ 3155
4005_904C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL57_4)	32	R/W	See section	55.5.61/ 3156
4005_9050	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL57_5)	32	R/W	See section	55.5.62/ 3158
4005_9054	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL57_6)	32	R/W	See section	55.5.63/ 3159
4005_9058	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL57_7)	32	R/W	See section	55.5.64/ 3160
4005_905C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL57_8)	32	R/W	See section	55.5.65/ 3161
4005_9060	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL57_9)	32	R/W	See section	55.5.66/ 3161
4005_9080	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL58_1)	32	R/W	See section	55.5.58/ 3154
4005_9084	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL58_2)	32	R/W	See section	55.5.59/ 3155
4005_9088	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL58_3)	32	R/W	See section	55.5.60/ 3155
4005_908C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL58_4)	32	R/W	See section	55.5.61/ 3156
4005_9090	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL58_5)	32	R/W	See section	55.5.62/ 3158
4005_9094	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL58_6)	32	R/W	See section	55.5.63/ 3159
4005_9098	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL58_7)	32	R/W	See section	55.5.64/ 3160
4005_909C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL58_8)	32	R/W	See section	55.5.65/ 3161
4005_90A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL58_9)	32	R/W	See section	55.5.66/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_90C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL59_1)	32	R/W	See section	55.5.58/ 3154
4005_90C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL59_2)	32	R/W	See section	55.5.59/ 3155
4005_90C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL59_3)	32	R/W	See section	55.5.60/ 3155
4005_90CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL59_4)	32	R/W	See section	55.5.61/ 3156
4005_90D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL59_5)	32	R/W	See section	55.5.62/ 3158
4005_90D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL59_6)	32	R/W	See section	55.5.63/ 3159
4005_90D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL59_7)	32	R/W	See section	55.5.64/ 3160
4005_90DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL59_8)	32	R/W	See section	55.5.65/ 3161
4005_90E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL59_9)	32	R/W	See section	55.5.66/ 3161
4005_9100	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL60_1)	32	R/W	See section	55.5.58/ 3154
4005_9104	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL60_2)	32	R/W	See section	55.5.59/ 3155
4005_9108	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL60_3)	32	R/W	See section	55.5.60/ 3155
4005_910C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL60_4)	32	R/W	See section	55.5.61/ 3156
4005_9110	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL60_5)	32	R/W	See section	55.5.62/ 3158
4005_9114	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL60_6)	32	R/W	See section	55.5.63/ 3159
4005_9118	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL60_7)	32	R/W	See section	55.5.64/ 3160
4005_911C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL60_8)	32	R/W	See section	55.5.65/ 3161
4005_9120	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL60_9)	32	R/W	See section	55.5.66/ 3161
4005_9140	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL61_1)	32	R/W	See section	55.5.58/ 3154
4005_9144	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL61_2)	32	R/W	See section	55.5.59/ 3155
4005_9148	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL61_3)	32	R/W	See section	55.5.60/ 3155
4005_914C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL61_4)	32	R/W	See section	55.5.61/ 3156

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_9150	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL61_5)	32	R/W	See section	55.5.62/ 3158
4005_9154	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL61_6)	32	R/W	See section	55.5.63/ 3159
4005_9158	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL61_7)	32	R/W	See section	55.5.64/ 3160
4005_915C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL61_8)	32	R/W	See section	55.5.65/ 3161
4005_9160	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL61_9)	32	R/W	See section	55.5.66/ 3161
4005_9180	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL62_1)	32	R/W	See section	55.5.58/ 3154
4005_9184	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL62_2)	32	R/W	See section	55.5.59/ 3155
4005_9188	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL62_3)	32	R/W	See section	55.5.60/ 3155
4005_918C	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL62_4)	32	R/W	See section	55.5.61/ 3156
4005_9190	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL62_5)	32	R/W	See section	55.5.62/ 3158
4005_9194	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL62_6)	32	R/W	See section	55.5.63/ 3159
4005_9198	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL62_7)	32	R/W	See section	55.5.64/ 3160
4005_919C	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL62_8)	32	R/W	See section	55.5.65/ 3161
4005_91A0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL62_9)	32	R/W	See section	55.5.66/ 3161
4005_91C0	Control Descriptor Ln_0 Register (DCU0_CTRLDESCL63_1)	32	R/W	See section	55.5.58/ 3154
4005_91C4	Control Descriptor Ln_1 Register (DCU0_CTRLDESCL63_2)	32	R/W	See section	55.5.59/ 3155
4005_91C8	Control Descriptor Ln_2 Register (DCU0_CTRLDESCL63_3)	32	R/W	See section	55.5.60/ 3155
4005_91CC	Control Descriptor Ln_3 Register (DCU0_CTRLDESCL63_4)	32	R/W	See section	55.5.61/ 3156
4005_91D0	Control Descriptor Ln_4 Register (DCU0_CTRLDESCL63_5)	32	R/W	See section	55.5.62/ 3158
4005_91D4	Control Descriptor Ln_5 Register (DCU0_CTRLDESCL63_6)	32	R/W	See section	55.5.63/ 3159
4005_91D8	Control Descriptor Ln_6 Register (DCU0_CTRLDESCL63_7)	32	R/W	See section	55.5.64/ 3160
4005_91DC	Control Descriptor Ln_7 Register (DCU0_CTRLDESCL63_8)	32	R/W	See section	55.5.65/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_91E0	Control Descriptor Ln_8 Register (DCU0_CTRLDESCL63_9)	32	R/W	See section	55.5.66/ 3161
400D_8000	Control Descriptor Cursor 1 Register (DCU1_CTRLDESCCURSOR1)	32	R/W	0000_0000h	55.5.1/3101
400D_8004	Control Descriptor Cursor 2 Register (DCU1_CTRLDESCCURSOR2)	32	R/W	0000_0000h	55.5.2/3101
400D_8008	Control Descriptor Cursor 3 Register (DCU1_CTRLDESCCURSOR3)	32	R/W	0000_0000h	55.5.3/3102
400D_800C	Control Descriptor Cursor 4 Register (DCU1_CTRLDESCCURSOR4)	32	R/W	0000_0000h	55.5.4/3103
400D_8010	DCU4 Mode Register (DCU1_DCU_MODE)	32	R/W	0000_8000h	55.5.5/3104
400D_8014	Background Register (DCU1_BGND)	32	R/W	0000_0000h	55.5.6/3107
400D_8018	Display Size Register (DCU1_DISP_SIZE)	32	R/W	0000_0000h	55.5.7/3107
400D_801C	Horizontal Sync Parameter Register (DCU1_HSYN_PARA)	32	R/W	00C0_1803h	55.5.8/3108
400D_8020	Vertical Sync Parameter Register (DCU1_VSYN_PARA)	32	R/W	00C0_1803h	55.5.9/3109
400D_8024	Synchronize Polarity Register (DCU1_SYNPOL)	32	R/W	0000_0000h	55.5.10/ 3110
400D_8028	Threshold Register (DCU1_THRESHOLD)	32	R/W	0000_780Ah	55.5.11/ 3112
400D_802C	Interrupt Status Register (DCU1_INT_STATUS)	32	R/W	0000_0000h	55.5.12/ 3112
400D_8030	Interrupt Mask Register (DCU1_INT_MASK)	32	R/W	FCFF_5FFFh	55.5.13/ 3114
400D_8034	COLBAR_1 Register (DCU1_COLBAR_1)	32	R/W	FF00_0000h	55.5.14/ 3117
400D_8038	COLBAR_2 Register (DCU1_COLBAR_2)	32	R/W	FF00_00FFh	55.5.15/ 3118
400D_803C	COLBAR_3 Register (DCU1_COLBAR_3)	32	R/W	FF00_FFFFh	55.5.16/ 3118
400D_8040	COLBAR_4 Register (DCU1_COLBAR_4)	32	R/W	FF00_FF00h	55.5.17/ 3119
400D_8044	COLBAR_5 Register (DCU1_COLBAR_5)	32	R/W	FFFF_FF00h	55.5.18/ 3120
400D_8048	COLBAR_6 Register (DCU1_COLBAR_6)	32	R/W	FFFF_0000h	55.5.19/ 3120
400D_804C	COLBAR_7 Register (DCU1_COLBAR_7)	32	R/W	FFFF_00FFh	55.5.20/ 3121
400D_8050	COLBAR_8 Register (DCU1_COLBAR_8)	32	R/W	FFFF_FFFFh	55.5.21/ 3122
400D_8054	Divide Ratio Register (DCU1_DIV_RATIO)	32	R/W	0000_0000h	55.5.22/ 3122
400D_8058	Sign Calculation 1 Register (DCU1_SIGN_CALC_1)	32	R/W	0000_0000h	55.5.23/ 3123

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_805C	Sign Calculation 2 Register (DCU1_SIGN_CALC_2)	32	R/W	0000_0000h	55.5.24/ 3123
400D_8060	CRC Value Register (DCU1_CRC_VAL)	32	R/W	0000_0000h	55.5.25/ 3124
400D_8064	PDI Status Register (DCU1_PDI_STATUS)	32	R/W	0000_0000h	55.5.26/ 3125
400D_8068	PDI Status Mask Register (DCU1_PDI_STA_MSK)	32	R/W	0000_03FFh	55.5.27/ 3126
400D_806C	Parameter Error Status 1 Register (DCU1_PARR_ERR_STATUS1)	32	R/W	0000_0000h	55.5.28/ 3127
400D_8070	Parameter Error Status 2 Register (DCU1_PARR_ERR_STATUS2)	32	R/W	0000_0000h	55.5.29/ 3128
400D_807C	Parameter Error Status 3 Register (DCU1_PARR_ERR_STATUS3)	32	R/W	0000_0000h	55.5.30/ 3129
400D_8080	Mask Parameter Error Status 1 Register (DCU1_MASK_PARR_ERR_STATUS1)	32	R/W	FFFF_FFFFh	55.5.31/ 3130
400D_8084	Mask Parameter Error Status 2 Register (DCU1_MASK_PARR_ERR_STATUS2)	32	R/W	FFFF_FFFFh	55.5.32/ 3130
400D_8090	Mask Parameter Error Status 3 Register (DCU1_MASK_PARR_ERR_STATUS3)	32	R/W	0007_FFFFh	55.5.33/ 3131
400D_8094	Threshold Input 1 Register (DCU1_THRESHOLD_INP_BUF_1)	32	R/W	7F00_7F00h	55.5.34/ 3132
400D_8098	Threshold Input 2 Register (DCU1_THRESHOLD_INP_BUF_2)	32	R/W	7F00_7F00h	55.5.35/ 3133
400D_809C	Threshold Input 3 Register (DCU1_THRESHOLD_INP_BUF_3)	32	R/W	7F00_7F00h	55.5.36/ 3134
400D_80A0	LUMA Component Register (DCU1_LUMA_COMP)	32	R/W	9512_A254h	55.5.37/ 3135
400D_80A4	Red Chroma Components Register (DCU1_CHROMA_RED)	32	R/W	0331_0000h	55.5.38/ 3136
400D_80A8	Green Chroma Components Register (DCU1_CHROMA_GREEN)	32	R/W	0660_0F38h	55.5.39/ 3136
400D_80AC	Blue Chroma Components Register (DCU1_CHROMA_BLUE)	32	R/W	0000_0409h	55.5.40/ 3137
400D_80B0	CRC Position Register (DCU1_CRC_POS)	32	R/W	0000_0000h	55.5.41/ 3138
400D_80B4	Layer Interpolation Enable Register (DCU1_LYR_INTPOL_EN)	32	R/W	0000_0409h	55.5.42/ 3138
400D_80B8	Layer Luminance Component Register (DCU1_LYR_LUMA_COMP)	32	R/W	9512_A254h	55.5.43/ 3139
400D_80BC	Layer Chroma Red Register (DCU1_LYR_CHRM_RED)	32	R/W	0331_0000h	55.5.44/ 3140
400D_80C0	Layer Chroma Green Register (DCU1_LYR_CHRM_GRN)	32	R/W	0660_0F38h	55.5.45/ 3140

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_80C4	Layer Chroma Blue Register (DCU1_LYR_CHRM_BLUE)	32	R/W	0000_0409h	55.5.46/ 3141
400D_80C8	Compression Image Size Register (DCU1_COMP_IMSIZE)	32	R/W	0000_0409h	55.5.47/ 3142
400D_80CC	Update Mode Register (DCU1_UPDATE_MODE)	32	R/W	0000_0000h	55.5.48/ 3142
400D_80D0	Underrun Register (DCU1_UNDERRUN)	32	R/W	0000_0000h	55.5.49/ 3143
400D_8100	Global Protection Register (DCU1_GLBL_PROTECT)	32	R/W	0000_0000h	55.5.50/ 3144
400D_8104	Soft Lock Bit Layer 0 Register (DCU1_SFT_LCK_BIT_L0)	32	R/W	0000_0000h	55.5.51/ 3145
400D_8108	Soft Lock Bit Layer 1 Register (DCU1_SFT_LCK_BIT_L1)	32	R/W	0000_0000h	55.5.52/ 3147
400D_810C	Soft Lock Display Size Register (DCU1_SFT_LCK_DISP_SIZE)	32	R/W	0000_0000h	55.5.53/ 3149
400D_8110	Soft Lock Hsync/Vsync Parameter Register (DCU1_SFT_LCK_HS_VS_PARA)	32	R/W	0000_0000h	55.5.54/ 3150
400D_8114	Soft Lock POL Register (DCU1_SFT_LCK_POL)	32	R/W	0000_0000h	55.5.55/ 3151
400D_8118	Soft Lock L0 Transparency Register (DCU1_SFT_LCK_L0_TRANSP)	32	R/W	0000_0000h	55.5.56/ 3152
400D_811C	Soft Lock L1 Transparency Register (DCU1_SFT_LCK_L1_TRANSP)	32	R/W	0000_0000h	55.5.57/ 3153
400D_8200	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL0_1)	32	R/W	See section	55.5.58/ 3154
400D_8204	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL0_2)	32	R/W	See section	55.5.59/ 3155
400D_8208	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL0_3)	32	R/W	See section	55.5.60/ 3155
400D_820C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL0_4)	32	R/W	See section	55.5.61/ 3156
400D_8210	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL0_5)	32	R/W	See section	55.5.62/ 3158
400D_8214	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL0_6)	32	R/W	See section	55.5.63/ 3159
400D_8218	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL0_7)	32	R/W	See section	55.5.64/ 3160
400D_821C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL0_8)	32	R/W	See section	55.5.65/ 3161
400D_8220	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL0_9)	32	R/W	See section	55.5.66/ 3161
400D_8240	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL1_1)	32	R/W	See section	55.5.58/ 3154

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8244	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL1_2)	32	R/W	See section	55.5.59/ 3155
400D_8248	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL1_3)	32	R/W	See section	55.5.60/ 3155
400D_824C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL1_4)	32	R/W	See section	55.5.61/ 3156
400D_8250	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL1_5)	32	R/W	See section	55.5.62/ 3158
400D_8254	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL1_6)	32	R/W	See section	55.5.63/ 3159
400D_8258	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL1_7)	32	R/W	See section	55.5.64/ 3160
400D_825C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL1_8)	32	R/W	See section	55.5.65/ 3161
400D_8260	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL1_9)	32	R/W	See section	55.5.66/ 3161
400D_8280	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL2_1)	32	R/W	See section	55.5.58/ 3154
400D_8284	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL2_2)	32	R/W	See section	55.5.59/ 3155
400D_8288	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL2_3)	32	R/W	See section	55.5.60/ 3155
400D_828C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL2_4)	32	R/W	See section	55.5.61/ 3156
400D_8290	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL2_5)	32	R/W	See section	55.5.62/ 3158
400D_8294	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL2_6)	32	R/W	See section	55.5.63/ 3159
400D_8298	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL2_7)	32	R/W	See section	55.5.64/ 3160
400D_829C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL2_8)	32	R/W	See section	55.5.65/ 3161
400D_82A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL2_9)	32	R/W	See section	55.5.66/ 3161
400D_82C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL3_1)	32	R/W	See section	55.5.58/ 3154
400D_82C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL3_2)	32	R/W	See section	55.5.59/ 3155
400D_82C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL3_3)	32	R/W	See section	55.5.60/ 3155
400D_82CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL3_4)	32	R/W	See section	55.5.61/ 3156
400D_82D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL3_5)	32	R/W	See section	55.5.62/ 3158

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_82D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL3_6)	32	R/W	See section	55.5.63/ 3159
400D_82D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL3_7)	32	R/W	See section	55.5.64/ 3160
400D_82DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL3_8)	32	R/W	See section	55.5.65/ 3161
400D_82E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL3_9)	32	R/W	See section	55.5.66/ 3161
400D_8300	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL4_1)	32	R/W	See section	55.5.58/ 3154
400D_8304	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL4_2)	32	R/W	See section	55.5.59/ 3155
400D_8308	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL4_3)	32	R/W	See section	55.5.60/ 3155
400D_830C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL4_4)	32	R/W	See section	55.5.61/ 3156
400D_8310	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL4_5)	32	R/W	See section	55.5.62/ 3158
400D_8314	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL4_6)	32	R/W	See section	55.5.63/ 3159
400D_8318	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL4_7)	32	R/W	See section	55.5.64/ 3160
400D_831C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL4_8)	32	R/W	See section	55.5.65/ 3161
400D_8320	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL4_9)	32	R/W	See section	55.5.66/ 3161
400D_8340	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL5_1)	32	R/W	See section	55.5.58/ 3154
400D_8344	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL5_2)	32	R/W	See section	55.5.59/ 3155
400D_8348	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL5_3)	32	R/W	See section	55.5.60/ 3155
400D_834C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL5_4)	32	R/W	See section	55.5.61/ 3156
400D_8350	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL5_5)	32	R/W	See section	55.5.62/ 3158
400D_8354	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL5_6)	32	R/W	See section	55.5.63/ 3159
400D_8358	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL5_7)	32	R/W	See section	55.5.64/ 3160
400D_835C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL5_8)	32	R/W	See section	55.5.65/ 3161
400D_8360	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL5_9)	32	R/W	See section	55.5.66/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8380	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL6_1)	32	R/W	See section	55.5.58/ 3154
400D_8384	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL6_2)	32	R/W	See section	55.5.59/ 3155
400D_8388	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL6_3)	32	R/W	See section	55.5.60/ 3155
400D_838C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL6_4)	32	R/W	See section	55.5.61/ 3156
400D_8390	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL6_5)	32	R/W	See section	55.5.62/ 3158
400D_8394	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL6_6)	32	R/W	See section	55.5.63/ 3159
400D_8398	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL6_7)	32	R/W	See section	55.5.64/ 3160
400D_839C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL6_8)	32	R/W	See section	55.5.65/ 3161
400D_83A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL6_9)	32	R/W	See section	55.5.66/ 3161
400D_83C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL7_1)	32	R/W	See section	55.5.58/ 3154
400D_83C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL7_2)	32	R/W	See section	55.5.59/ 3155
400D_83C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL7_3)	32	R/W	See section	55.5.60/ 3155
400D_83CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL7_4)	32	R/W	See section	55.5.61/ 3156
400D_83D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL7_5)	32	R/W	See section	55.5.62/ 3158
400D_83D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL7_6)	32	R/W	See section	55.5.63/ 3159
400D_83D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL7_7)	32	R/W	See section	55.5.64/ 3160
400D_83DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL7_8)	32	R/W	See section	55.5.65/ 3161
400D_83E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL7_9)	32	R/W	See section	55.5.66/ 3161
400D_8400	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL8_1)	32	R/W	See section	55.5.58/ 3154
400D_8404	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL8_2)	32	R/W	See section	55.5.59/ 3155
400D_8408	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL8_3)	32	R/W	See section	55.5.60/ 3155
400D_840C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL8_4)	32	R/W	See section	55.5.61/ 3156

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8410	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL8_5)	32	R/W	See section	55.5.62/ 3158
400D_8414	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL8_6)	32	R/W	See section	55.5.63/ 3159
400D_8418	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL8_7)	32	R/W	See section	55.5.64/ 3160
400D_841C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL8_8)	32	R/W	See section	55.5.65/ 3161
400D_8420	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL8_9)	32	R/W	See section	55.5.66/ 3161
400D_8440	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL9_1)	32	R/W	See section	55.5.58/ 3154
400D_8444	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL9_2)	32	R/W	See section	55.5.59/ 3155
400D_8448	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL9_3)	32	R/W	See section	55.5.60/ 3155
400D_844C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL9_4)	32	R/W	See section	55.5.61/ 3156
400D_8450	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL9_5)	32	R/W	See section	55.5.62/ 3158
400D_8454	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL9_6)	32	R/W	See section	55.5.63/ 3159
400D_8458	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL9_7)	32	R/W	See section	55.5.64/ 3160
400D_845C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL9_8)	32	R/W	See section	55.5.65/ 3161
400D_8460	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL9_9)	32	R/W	See section	55.5.66/ 3161
400D_8480	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL10_1)	32	R/W	See section	55.5.58/ 3154
400D_8484	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL10_2)	32	R/W	See section	55.5.59/ 3155
400D_8488	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL10_3)	32	R/W	See section	55.5.60/ 3155
400D_848C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL10_4)	32	R/W	See section	55.5.61/ 3156
400D_8490	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL10_5)	32	R/W	See section	55.5.62/ 3158
400D_8494	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL10_6)	32	R/W	See section	55.5.63/ 3159
400D_8498	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL10_7)	32	R/W	See section	55.5.64/ 3160
400D_849C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL10_8)	32	R/W	See section	55.5.65/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_84A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL10_9)	32	R/W	See section	55.5.66/ 3161
400D_84C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL11_1)	32	R/W	See section	55.5.58/ 3154
400D_84C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL11_2)	32	R/W	See section	55.5.59/ 3155
400D_84C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL11_3)	32	R/W	See section	55.5.60/ 3155
400D_84CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL11_4)	32	R/W	See section	55.5.61/ 3156
400D_84D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL11_5)	32	R/W	See section	55.5.62/ 3158
400D_84D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL11_6)	32	R/W	See section	55.5.63/ 3159
400D_84D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL11_7)	32	R/W	See section	55.5.64/ 3160
400D_84DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL11_8)	32	R/W	See section	55.5.65/ 3161
400D_84E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL11_9)	32	R/W	See section	55.5.66/ 3161
400D_8500	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL12_1)	32	R/W	See section	55.5.58/ 3154
400D_8504	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL12_2)	32	R/W	See section	55.5.59/ 3155
400D_8508	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL12_3)	32	R/W	See section	55.5.60/ 3155
400D_850C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL12_4)	32	R/W	See section	55.5.61/ 3156
400D_8510	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL12_5)	32	R/W	See section	55.5.62/ 3158
400D_8514	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL12_6)	32	R/W	See section	55.5.63/ 3159
400D_8518	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL12_7)	32	R/W	See section	55.5.64/ 3160
400D_851C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL12_8)	32	R/W	See section	55.5.65/ 3161
400D_8520	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL12_9)	32	R/W	See section	55.5.66/ 3161
400D_8540	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL13_1)	32	R/W	See section	55.5.58/ 3154
400D_8544	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL13_2)	32	R/W	See section	55.5.59/ 3155
400D_8548	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL13_3)	32	R/W	See section	55.5.60/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_854C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL13_4)	32	R/W	See section	55.5.61/ 3156
400D_8550	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL13_5)	32	R/W	See section	55.5.62/ 3158
400D_8554	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL13_6)	32	R/W	See section	55.5.63/ 3159
400D_8558	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL13_7)	32	R/W	See section	55.5.64/ 3160
400D_855C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL13_8)	32	R/W	See section	55.5.65/ 3161
400D_8560	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL13_9)	32	R/W	See section	55.5.66/ 3161
400D_8580	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL14_1)	32	R/W	See section	55.5.58/ 3154
400D_8584	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL14_2)	32	R/W	See section	55.5.59/ 3155
400D_8588	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL14_3)	32	R/W	See section	55.5.60/ 3155
400D_858C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL14_4)	32	R/W	See section	55.5.61/ 3156
400D_8590	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL14_5)	32	R/W	See section	55.5.62/ 3158
400D_8594	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL14_6)	32	R/W	See section	55.5.63/ 3159
400D_8598	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL14_7)	32	R/W	See section	55.5.64/ 3160
400D_859C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL14_8)	32	R/W	See section	55.5.65/ 3161
400D_85A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL14_9)	32	R/W	See section	55.5.66/ 3161
400D_85C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL15_1)	32	R/W	See section	55.5.58/ 3154
400D_85C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL15_2)	32	R/W	See section	55.5.59/ 3155
400D_85C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL15_3)	32	R/W	See section	55.5.60/ 3155
400D_85CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL15_4)	32	R/W	See section	55.5.61/ 3156
400D_85D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL15_5)	32	R/W	See section	55.5.62/ 3158
400D_85D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL15_6)	32	R/W	See section	55.5.63/ 3159
400D_85D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL15_7)	32	R/W	See section	55.5.64/ 3160

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_85DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL15_8)	32	R/W	See section	55.5.65/ 3161
400D_85E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL15_9)	32	R/W	See section	55.5.66/ 3161
400D_8600	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL16_1)	32	R/W	See section	55.5.58/ 3154
400D_8604	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL16_2)	32	R/W	See section	55.5.59/ 3155
400D_8608	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL16_3)	32	R/W	See section	55.5.60/ 3155
400D_860C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL16_4)	32	R/W	See section	55.5.61/ 3156
400D_8610	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL16_5)	32	R/W	See section	55.5.62/ 3158
400D_8614	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL16_6)	32	R/W	See section	55.5.63/ 3159
400D_8618	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL16_7)	32	R/W	See section	55.5.64/ 3160
400D_861C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL16_8)	32	R/W	See section	55.5.65/ 3161
400D_8620	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL16_9)	32	R/W	See section	55.5.66/ 3161
400D_8640	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL17_1)	32	R/W	See section	55.5.58/ 3154
400D_8644	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL17_2)	32	R/W	See section	55.5.59/ 3155
400D_8648	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL17_3)	32	R/W	See section	55.5.60/ 3155
400D_864C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL17_4)	32	R/W	See section	55.5.61/ 3156
400D_8650	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL17_5)	32	R/W	See section	55.5.62/ 3158
400D_8654	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL17_6)	32	R/W	See section	55.5.63/ 3159
400D_8658	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL17_7)	32	R/W	See section	55.5.64/ 3160
400D_865C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL17_8)	32	R/W	See section	55.5.65/ 3161
400D_8660	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL17_9)	32	R/W	See section	55.5.66/ 3161
400D_8680	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL18_1)	32	R/W	See section	55.5.58/ 3154
400D_8684	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL18_2)	32	R/W	See section	55.5.59/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8688	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL18_3)	32	R/W	See section	55.5.60/ 3155
400D_868C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL18_4)	32	R/W	See section	55.5.61/ 3156
400D_8690	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL18_5)	32	R/W	See section	55.5.62/ 3158
400D_8694	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL18_6)	32	R/W	See section	55.5.63/ 3159
400D_8698	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL18_7)	32	R/W	See section	55.5.64/ 3160
400D_869C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL18_8)	32	R/W	See section	55.5.65/ 3161
400D_86A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL18_9)	32	R/W	See section	55.5.66/ 3161
400D_86C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL19_1)	32	R/W	See section	55.5.58/ 3154
400D_86C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL19_2)	32	R/W	See section	55.5.59/ 3155
400D_86C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL19_3)	32	R/W	See section	55.5.60/ 3155
400D_86CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL19_4)	32	R/W	See section	55.5.61/ 3156
400D_86D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL19_5)	32	R/W	See section	55.5.62/ 3158
400D_86D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL19_6)	32	R/W	See section	55.5.63/ 3159
400D_86D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL19_7)	32	R/W	See section	55.5.64/ 3160
400D_86DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL19_8)	32	R/W	See section	55.5.65/ 3161
400D_86E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL19_9)	32	R/W	See section	55.5.66/ 3161
400D_8700	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL20_1)	32	R/W	See section	55.5.58/ 3154
400D_8704	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL20_2)	32	R/W	See section	55.5.59/ 3155
400D_8708	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL20_3)	32	R/W	See section	55.5.60/ 3155
400D_870C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL20_4)	32	R/W	See section	55.5.61/ 3156
400D_8710	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL20_5)	32	R/W	See section	55.5.62/ 3158
400D_8714	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL20_6)	32	R/W	See section	55.5.63/ 3159

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8718	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL20_7)	32	R/W	See section	55.5.64/ 3160
400D_871C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL20_8)	32	R/W	See section	55.5.65/ 3161
400D_8720	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL20_9)	32	R/W	See section	55.5.66/ 3161
400D_8740	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL21_1)	32	R/W	See section	55.5.58/ 3154
400D_8744	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL21_2)	32	R/W	See section	55.5.59/ 3155
400D_8748	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL21_3)	32	R/W	See section	55.5.60/ 3155
400D_874C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL21_4)	32	R/W	See section	55.5.61/ 3156
400D_8750	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL21_5)	32	R/W	See section	55.5.62/ 3158
400D_8754	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL21_6)	32	R/W	See section	55.5.63/ 3159
400D_8758	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL21_7)	32	R/W	See section	55.5.64/ 3160
400D_875C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL21_8)	32	R/W	See section	55.5.65/ 3161
400D_8760	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL21_9)	32	R/W	See section	55.5.66/ 3161
400D_8780	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL22_1)	32	R/W	See section	55.5.58/ 3154
400D_8784	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL22_2)	32	R/W	See section	55.5.59/ 3155
400D_8788	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL22_3)	32	R/W	See section	55.5.60/ 3155
400D_878C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL22_4)	32	R/W	See section	55.5.61/ 3156
400D_8790	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL22_5)	32	R/W	See section	55.5.62/ 3158
400D_8794	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL22_6)	32	R/W	See section	55.5.63/ 3159
400D_8798	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL22_7)	32	R/W	See section	55.5.64/ 3160
400D_879C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL22_8)	32	R/W	See section	55.5.65/ 3161
400D_87A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL22_9)	32	R/W	See section	55.5.66/ 3161
400D_87C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL23_1)	32	R/W	See section	55.5.58/ 3154

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_87C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL23_2)	32	R/W	See section	55.5.59/ 3155
400D_87C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL23_3)	32	R/W	See section	55.5.60/ 3155
400D_87CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL23_4)	32	R/W	See section	55.5.61/ 3156
400D_87D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL23_5)	32	R/W	See section	55.5.62/ 3158
400D_87D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL23_6)	32	R/W	See section	55.5.63/ 3159
400D_87D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL23_7)	32	R/W	See section	55.5.64/ 3160
400D_87DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL23_8)	32	R/W	See section	55.5.65/ 3161
400D_87E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL23_9)	32	R/W	See section	55.5.66/ 3161
400D_8800	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL24_1)	32	R/W	See section	55.5.58/ 3154
400D_8804	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL24_2)	32	R/W	See section	55.5.59/ 3155
400D_8808	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL24_3)	32	R/W	See section	55.5.60/ 3155
400D_880C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL24_4)	32	R/W	See section	55.5.61/ 3156
400D_8810	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL24_5)	32	R/W	See section	55.5.62/ 3158
400D_8814	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL24_6)	32	R/W	See section	55.5.63/ 3159
400D_8818	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL24_7)	32	R/W	See section	55.5.64/ 3160
400D_881C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL24_8)	32	R/W	See section	55.5.65/ 3161
400D_8820	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL24_9)	32	R/W	See section	55.5.66/ 3161
400D_8840	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL25_1)	32	R/W	See section	55.5.58/ 3154
400D_8844	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL25_2)	32	R/W	See section	55.5.59/ 3155
400D_8848	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL25_3)	32	R/W	See section	55.5.60/ 3155
400D_884C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL25_4)	32	R/W	See section	55.5.61/ 3156
400D_8850	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL25_5)	32	R/W	See section	55.5.62/ 3158

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8854	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL25_6)	32	R/W	See section	55.5.63/ 3159
400D_8858	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL25_7)	32	R/W	See section	55.5.64/ 3160
400D_885C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL25_8)	32	R/W	See section	55.5.65/ 3161
400D_8860	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL25_9)	32	R/W	See section	55.5.66/ 3161
400D_8880	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL26_1)	32	R/W	See section	55.5.58/ 3154
400D_8884	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL26_2)	32	R/W	See section	55.5.59/ 3155
400D_8888	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL26_3)	32	R/W	See section	55.5.60/ 3155
400D_888C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL26_4)	32	R/W	See section	55.5.61/ 3156
400D_8890	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL26_5)	32	R/W	See section	55.5.62/ 3158
400D_8894	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL26_6)	32	R/W	See section	55.5.63/ 3159
400D_8898	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL26_7)	32	R/W	See section	55.5.64/ 3160
400D_889C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL26_8)	32	R/W	See section	55.5.65/ 3161
400D_88A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL26_9)	32	R/W	See section	55.5.66/ 3161
400D_88C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL27_1)	32	R/W	See section	55.5.58/ 3154
400D_88C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL27_2)	32	R/W	See section	55.5.59/ 3155
400D_88C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL27_3)	32	R/W	See section	55.5.60/ 3155
400D_88CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL27_4)	32	R/W	See section	55.5.61/ 3156
400D_88D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL27_5)	32	R/W	See section	55.5.62/ 3158
400D_88D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL27_6)	32	R/W	See section	55.5.63/ 3159
400D_88D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL27_7)	32	R/W	See section	55.5.64/ 3160
400D_88DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL27_8)	32	R/W	See section	55.5.65/ 3161
400D_88E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL27_9)	32	R/W	See section	55.5.66/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8900	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL28_1)	32	R/W	See section	55.5.58/ 3154
400D_8904	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL28_2)	32	R/W	See section	55.5.59/ 3155
400D_8908	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL28_3)	32	R/W	See section	55.5.60/ 3155
400D_890C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL28_4)	32	R/W	See section	55.5.61/ 3156
400D_8910	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL28_5)	32	R/W	See section	55.5.62/ 3158
400D_8914	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL28_6)	32	R/W	See section	55.5.63/ 3159
400D_8918	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL28_7)	32	R/W	See section	55.5.64/ 3160
400D_891C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL28_8)	32	R/W	See section	55.5.65/ 3161
400D_8920	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL28_9)	32	R/W	See section	55.5.66/ 3161
400D_8940	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL29_1)	32	R/W	See section	55.5.58/ 3154
400D_8944	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL29_2)	32	R/W	See section	55.5.59/ 3155
400D_8948	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL29_3)	32	R/W	See section	55.5.60/ 3155
400D_894C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL29_4)	32	R/W	See section	55.5.61/ 3156
400D_8950	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL29_5)	32	R/W	See section	55.5.62/ 3158
400D_8954	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL29_6)	32	R/W	See section	55.5.63/ 3159
400D_8958	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL29_7)	32	R/W	See section	55.5.64/ 3160
400D_895C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL29_8)	32	R/W	See section	55.5.65/ 3161
400D_8960	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL29_9)	32	R/W	See section	55.5.66/ 3161
400D_8980	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL30_1)	32	R/W	See section	55.5.58/ 3154
400D_8984	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL30_2)	32	R/W	See section	55.5.59/ 3155
400D_8988	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL30_3)	32	R/W	See section	55.5.60/ 3155
400D_898C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL30_4)	32	R/W	See section	55.5.61/ 3156

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8990	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL30_5)	32	R/W	See section	55.5.62/ 3158
400D_8994	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL30_6)	32	R/W	See section	55.5.63/ 3159
400D_8998	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL30_7)	32	R/W	See section	55.5.64/ 3160
400D_899C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL30_8)	32	R/W	See section	55.5.65/ 3161
400D_89A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL30_9)	32	R/W	See section	55.5.66/ 3161
400D_89C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL31_1)	32	R/W	See section	55.5.58/ 3154
400D_89C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL31_2)	32	R/W	See section	55.5.59/ 3155
400D_89C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL31_3)	32	R/W	See section	55.5.60/ 3155
400D_89CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL31_4)	32	R/W	See section	55.5.61/ 3156
400D_89D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL31_5)	32	R/W	See section	55.5.62/ 3158
400D_89D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL31_6)	32	R/W	See section	55.5.63/ 3159
400D_89D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL31_7)	32	R/W	See section	55.5.64/ 3160
400D_89DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL31_8)	32	R/W	See section	55.5.65/ 3161
400D_89E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL31_9)	32	R/W	See section	55.5.66/ 3161
400D_8A00	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL32_1)	32	R/W	See section	55.5.58/ 3154
400D_8A04	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL32_2)	32	R/W	See section	55.5.59/ 3155
400D_8A08	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL32_3)	32	R/W	See section	55.5.60/ 3155
400D_8A0C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL32_4)	32	R/W	See section	55.5.61/ 3156
400D_8A10	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL32_5)	32	R/W	See section	55.5.62/ 3158
400D_8A14	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL32_6)	32	R/W	See section	55.5.63/ 3159
400D_8A18	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL32_7)	32	R/W	See section	55.5.64/ 3160
400D_8A1C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL32_8)	32	R/W	See section	55.5.65/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8A20	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL32_9)	32	R/W	See section	55.5.66/ 3161
400D_8A40	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL33_1)	32	R/W	See section	55.5.58/ 3154
400D_8A44	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL33_2)	32	R/W	See section	55.5.59/ 3155
400D_8A48	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL33_3)	32	R/W	See section	55.5.60/ 3155
400D_8A4C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL33_4)	32	R/W	See section	55.5.61/ 3156
400D_8A50	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL33_5)	32	R/W	See section	55.5.62/ 3158
400D_8A54	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL33_6)	32	R/W	See section	55.5.63/ 3159
400D_8A58	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL33_7)	32	R/W	See section	55.5.64/ 3160
400D_8A5C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL33_8)	32	R/W	See section	55.5.65/ 3161
400D_8A60	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL33_9)	32	R/W	See section	55.5.66/ 3161
400D_8A80	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL34_1)	32	R/W	See section	55.5.58/ 3154
400D_8A84	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL34_2)	32	R/W	See section	55.5.59/ 3155
400D_8A88	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL34_3)	32	R/W	See section	55.5.60/ 3155
400D_8A8C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL34_4)	32	R/W	See section	55.5.61/ 3156
400D_8A90	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL34_5)	32	R/W	See section	55.5.62/ 3158
400D_8A94	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL34_6)	32	R/W	See section	55.5.63/ 3159
400D_8A98	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL34_7)	32	R/W	See section	55.5.64/ 3160
400D_8A9C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL34_8)	32	R/W	See section	55.5.65/ 3161
400D_8AA0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL34_9)	32	R/W	See section	55.5.66/ 3161
400D_8AC0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL35_1)	32	R/W	See section	55.5.58/ 3154
400D_8AC4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL35_2)	32	R/W	See section	55.5.59/ 3155
400D_8AC8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL35_3)	32	R/W	See section	55.5.60/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8ACC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL35_4)	32	R/W	See section	55.5.61/ 3156
400D_8AD0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL35_5)	32	R/W	See section	55.5.62/ 3158
400D_8AD4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL35_6)	32	R/W	See section	55.5.63/ 3159
400D_8AD8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL35_7)	32	R/W	See section	55.5.64/ 3160
400D_8ADC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL35_8)	32	R/W	See section	55.5.65/ 3161
400D_8AE0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL35_9)	32	R/W	See section	55.5.66/ 3161
400D_8B00	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL36_1)	32	R/W	See section	55.5.58/ 3154
400D_8B04	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL36_2)	32	R/W	See section	55.5.59/ 3155
400D_8B08	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL36_3)	32	R/W	See section	55.5.60/ 3155
400D_8B0C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL36_4)	32	R/W	See section	55.5.61/ 3156
400D_8B10	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL36_5)	32	R/W	See section	55.5.62/ 3158
400D_8B14	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL36_6)	32	R/W	See section	55.5.63/ 3159
400D_8B18	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL36_7)	32	R/W	See section	55.5.64/ 3160
400D_8B1C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL36_8)	32	R/W	See section	55.5.65/ 3161
400D_8B20	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL36_9)	32	R/W	See section	55.5.66/ 3161
400D_8B40	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL37_1)	32	R/W	See section	55.5.58/ 3154
400D_8B44	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL37_2)	32	R/W	See section	55.5.59/ 3155
400D_8B48	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL37_3)	32	R/W	See section	55.5.60/ 3155
400D_8B4C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL37_4)	32	R/W	See section	55.5.61/ 3156
400D_8B50	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL37_5)	32	R/W	See section	55.5.62/ 3158
400D_8B54	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL37_6)	32	R/W	See section	55.5.63/ 3159
400D_8B58	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL37_7)	32	R/W	See section	55.5.64/ 3160

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8B5C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL37_8)	32	R/W	See section	55.5.65/ 3161
400D_8B60	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL37_9)	32	R/W	See section	55.5.66/ 3161
400D_8B80	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL38_1)	32	R/W	See section	55.5.58/ 3154
400D_8B84	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL38_2)	32	R/W	See section	55.5.59/ 3155
400D_8B88	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL38_3)	32	R/W	See section	55.5.60/ 3155
400D_8B8C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL38_4)	32	R/W	See section	55.5.61/ 3156
400D_8B90	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL38_5)	32	R/W	See section	55.5.62/ 3158
400D_8B94	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL38_6)	32	R/W	See section	55.5.63/ 3159
400D_8B98	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL38_7)	32	R/W	See section	55.5.64/ 3160
400D_8B9C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL38_8)	32	R/W	See section	55.5.65/ 3161
400D_8BA0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL38_9)	32	R/W	See section	55.5.66/ 3161
400D_8BC0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL39_1)	32	R/W	See section	55.5.58/ 3154
400D_8BC4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL39_2)	32	R/W	See section	55.5.59/ 3155
400D_8BC8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL39_3)	32	R/W	See section	55.5.60/ 3155
400D_8BCC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL39_4)	32	R/W	See section	55.5.61/ 3156
400D_8BD0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL39_5)	32	R/W	See section	55.5.62/ 3158
400D_8BD4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL39_6)	32	R/W	See section	55.5.63/ 3159
400D_8BD8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL39_7)	32	R/W	See section	55.5.64/ 3160
400D_8BDC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL39_8)	32	R/W	See section	55.5.65/ 3161
400D_8BE0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL39_9)	32	R/W	See section	55.5.66/ 3161
400D_8C00	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL40_1)	32	R/W	See section	55.5.58/ 3154
400D_8C04	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL40_2)	32	R/W	See section	55.5.59/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8C08	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL40_3)	32	R/W	See section	55.5.60/ 3155
400D_8C0C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL40_4)	32	R/W	See section	55.5.61/ 3156
400D_8C10	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL40_5)	32	R/W	See section	55.5.62/ 3158
400D_8C14	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL40_6)	32	R/W	See section	55.5.63/ 3159
400D_8C18	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL40_7)	32	R/W	See section	55.5.64/ 3160
400D_8C1C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL40_8)	32	R/W	See section	55.5.65/ 3161
400D_8C20	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL40_9)	32	R/W	See section	55.5.66/ 3161
400D_8C40	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL41_1)	32	R/W	See section	55.5.58/ 3154
400D_8C44	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL41_2)	32	R/W	See section	55.5.59/ 3155
400D_8C48	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL41_3)	32	R/W	See section	55.5.60/ 3155
400D_8C4C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL41_4)	32	R/W	See section	55.5.61/ 3156
400D_8C50	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL41_5)	32	R/W	See section	55.5.62/ 3158
400D_8C54	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL41_6)	32	R/W	See section	55.5.63/ 3159
400D_8C58	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL41_7)	32	R/W	See section	55.5.64/ 3160
400D_8C5C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL41_8)	32	R/W	See section	55.5.65/ 3161
400D_8C60	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL41_9)	32	R/W	See section	55.5.66/ 3161
400D_8C80	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL42_1)	32	R/W	See section	55.5.58/ 3154
400D_8C84	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL42_2)	32	R/W	See section	55.5.59/ 3155
400D_8C88	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL42_3)	32	R/W	See section	55.5.60/ 3155
400D_8C8C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL42_4)	32	R/W	See section	55.5.61/ 3156
400D_8C90	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL42_5)	32	R/W	See section	55.5.62/ 3158
400D_8C94	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL42_6)	32	R/W	See section	55.5.63/ 3159

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8C98	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL42_7)	32	R/W	See section	55.5.64/ 3160
400D_8C9C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL42_8)	32	R/W	See section	55.5.65/ 3161
400D_8CA0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL42_9)	32	R/W	See section	55.5.66/ 3161
400D_8CC0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL43_1)	32	R/W	See section	55.5.58/ 3154
400D_8CC4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL43_2)	32	R/W	See section	55.5.59/ 3155
400D_8CC8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL43_3)	32	R/W	See section	55.5.60/ 3155
400D_8CCC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL43_4)	32	R/W	See section	55.5.61/ 3156
400D_8CD0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL43_5)	32	R/W	See section	55.5.62/ 3158
400D_8CD4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL43_6)	32	R/W	See section	55.5.63/ 3159
400D_8CD8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL43_7)	32	R/W	See section	55.5.64/ 3160
400D_8CDC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL43_8)	32	R/W	See section	55.5.65/ 3161
400D_8CE0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL43_9)	32	R/W	See section	55.5.66/ 3161
400D_8D00	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL44_1)	32	R/W	See section	55.5.58/ 3154
400D_8D04	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL44_2)	32	R/W	See section	55.5.59/ 3155
400D_8D08	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL44_3)	32	R/W	See section	55.5.60/ 3155
400D_8D0C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL44_4)	32	R/W	See section	55.5.61/ 3156
400D_8D10	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL44_5)	32	R/W	See section	55.5.62/ 3158
400D_8D14	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL44_6)	32	R/W	See section	55.5.63/ 3159
400D_8D18	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL44_7)	32	R/W	See section	55.5.64/ 3160
400D_8D1C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL44_8)	32	R/W	See section	55.5.65/ 3161
400D_8D20	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL44_9)	32	R/W	See section	55.5.66/ 3161
400D_8D40	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL45_1)	32	R/W	See section	55.5.58/ 3154

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8D44	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL45_2)	32	R/W	See section	55.5.59/ 3155
400D_8D48	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL45_3)	32	R/W	See section	55.5.60/ 3155
400D_8D4C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL45_4)	32	R/W	See section	55.5.61/ 3156
400D_8D50	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL45_5)	32	R/W	See section	55.5.62/ 3158
400D_8D54	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL45_6)	32	R/W	See section	55.5.63/ 3159
400D_8D58	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL45_7)	32	R/W	See section	55.5.64/ 3160
400D_8D5C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL45_8)	32	R/W	See section	55.5.65/ 3161
400D_8D60	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL45_9)	32	R/W	See section	55.5.66/ 3161
400D_8D80	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL46_1)	32	R/W	See section	55.5.58/ 3154
400D_8D84	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL46_2)	32	R/W	See section	55.5.59/ 3155
400D_8D88	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL46_3)	32	R/W	See section	55.5.60/ 3155
400D_8D8C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL46_4)	32	R/W	See section	55.5.61/ 3156
400D_8D90	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL46_5)	32	R/W	See section	55.5.62/ 3158
400D_8D94	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL46_6)	32	R/W	See section	55.5.63/ 3159
400D_8D98	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL46_7)	32	R/W	See section	55.5.64/ 3160
400D_8D9C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL46_8)	32	R/W	See section	55.5.65/ 3161
400D_8DA0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL46_9)	32	R/W	See section	55.5.66/ 3161
400D_8DC0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL47_1)	32	R/W	See section	55.5.58/ 3154
400D_8DC4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL47_2)	32	R/W	See section	55.5.59/ 3155
400D_8DC8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL47_3)	32	R/W	See section	55.5.60/ 3155
400D_8DCC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL47_4)	32	R/W	See section	55.5.61/ 3156
400D_8DD0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL47_5)	32	R/W	See section	55.5.62/ 3158

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8DD4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL47_6)	32	R/W	See section	55.5.63/ 3159
400D_8DD8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL47_7)	32	R/W	See section	55.5.64/ 3160
400D_8DDC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL47_8)	32	R/W	See section	55.5.65/ 3161
400D_8DE0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL47_9)	32	R/W	See section	55.5.66/ 3161
400D_8E00	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL48_1)	32	R/W	See section	55.5.58/ 3154
400D_8E04	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL48_2)	32	R/W	See section	55.5.59/ 3155
400D_8E08	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL48_3)	32	R/W	See section	55.5.60/ 3155
400D_8E0C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL48_4)	32	R/W	See section	55.5.61/ 3156
400D_8E10	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL48_5)	32	R/W	See section	55.5.62/ 3158
400D_8E14	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL48_6)	32	R/W	See section	55.5.63/ 3159
400D_8E18	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL48_7)	32	R/W	See section	55.5.64/ 3160
400D_8E1C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL48_8)	32	R/W	See section	55.5.65/ 3161
400D_8E20	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL48_9)	32	R/W	See section	55.5.66/ 3161
400D_8E40	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL49_1)	32	R/W	See section	55.5.58/ 3154
400D_8E44	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL49_2)	32	R/W	See section	55.5.59/ 3155
400D_8E48	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL49_3)	32	R/W	See section	55.5.60/ 3155
400D_8E4C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL49_4)	32	R/W	See section	55.5.61/ 3156
400D_8E50	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL49_5)	32	R/W	See section	55.5.62/ 3158
400D_8E54	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL49_6)	32	R/W	See section	55.5.63/ 3159
400D_8E58	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL49_7)	32	R/W	See section	55.5.64/ 3160
400D_8E5C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL49_8)	32	R/W	See section	55.5.65/ 3161
400D_8E60	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL49_9)	32	R/W	See section	55.5.66/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8E80	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL50_1)	32	R/W	See section	55.5.58/ 3154
400D_8E84	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL50_2)	32	R/W	See section	55.5.59/ 3155
400D_8E88	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL50_3)	32	R/W	See section	55.5.60/ 3155
400D_8E8C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL50_4)	32	R/W	See section	55.5.61/ 3156
400D_8E90	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL50_5)	32	R/W	See section	55.5.62/ 3158
400D_8E94	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL50_6)	32	R/W	See section	55.5.63/ 3159
400D_8E98	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL50_7)	32	R/W	See section	55.5.64/ 3160
400D_8E9C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL50_8)	32	R/W	See section	55.5.65/ 3161
400D_8EA0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL50_9)	32	R/W	See section	55.5.66/ 3161
400D_8EC0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL51_1)	32	R/W	See section	55.5.58/ 3154
400D_8EC4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL51_2)	32	R/W	See section	55.5.59/ 3155
400D_8EC8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL51_3)	32	R/W	See section	55.5.60/ 3155
400D_8ECC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL51_4)	32	R/W	See section	55.5.61/ 3156
400D_8ED0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL51_5)	32	R/W	See section	55.5.62/ 3158
400D_8ED4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL51_6)	32	R/W	See section	55.5.63/ 3159
400D_8ED8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL51_7)	32	R/W	See section	55.5.64/ 3160
400D_8EDC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL51_8)	32	R/W	See section	55.5.65/ 3161
400D_8EE0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL51_9)	32	R/W	See section	55.5.66/ 3161
400D_8F00	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL52_1)	32	R/W	See section	55.5.58/ 3154
400D_8F04	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL52_2)	32	R/W	See section	55.5.59/ 3155
400D_8F08	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL52_3)	32	R/W	See section	55.5.60/ 3155
400D_8F0C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL52_4)	32	R/W	See section	55.5.61/ 3156

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8F10	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL52_5)	32	R/W	See section	55.5.62/ 3158
400D_8F14	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL52_6)	32	R/W	See section	55.5.63/ 3159
400D_8F18	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL52_7)	32	R/W	See section	55.5.64/ 3160
400D_8F1C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL52_8)	32	R/W	See section	55.5.65/ 3161
400D_8F20	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL52_9)	32	R/W	See section	55.5.66/ 3161
400D_8F40	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL53_1)	32	R/W	See section	55.5.58/ 3154
400D_8F44	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL53_2)	32	R/W	See section	55.5.59/ 3155
400D_8F48	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL53_3)	32	R/W	See section	55.5.60/ 3155
400D_8F4C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL53_4)	32	R/W	See section	55.5.61/ 3156
400D_8F50	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL53_5)	32	R/W	See section	55.5.62/ 3158
400D_8F54	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL53_6)	32	R/W	See section	55.5.63/ 3159
400D_8F58	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL53_7)	32	R/W	See section	55.5.64/ 3160
400D_8F5C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL53_8)	32	R/W	See section	55.5.65/ 3161
400D_8F60	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL53_9)	32	R/W	See section	55.5.66/ 3161
400D_8F80	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL54_1)	32	R/W	See section	55.5.58/ 3154
400D_8F84	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL54_2)	32	R/W	See section	55.5.59/ 3155
400D_8F88	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL54_3)	32	R/W	See section	55.5.60/ 3155
400D_8F8C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL54_4)	32	R/W	See section	55.5.61/ 3156
400D_8F90	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL54_5)	32	R/W	See section	55.5.62/ 3158
400D_8F94	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL54_6)	32	R/W	See section	55.5.63/ 3159
400D_8F98	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL54_7)	32	R/W	See section	55.5.64/ 3160
400D_8F9C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL54_8)	32	R/W	See section	55.5.65/ 3161

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_8FA0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL54_9)	32	R/W	See section	55.5.66/ 3161
400D_8FC0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL55_1)	32	R/W	See section	55.5.58/ 3154
400D_8FC4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL55_2)	32	R/W	See section	55.5.59/ 3155
400D_8FC8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL55_3)	32	R/W	See section	55.5.60/ 3155
400D_8FCC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL55_4)	32	R/W	See section	55.5.61/ 3156
400D_8FD0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL55_5)	32	R/W	See section	55.5.62/ 3158
400D_8FD4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL55_6)	32	R/W	See section	55.5.63/ 3159
400D_8FD8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL55_7)	32	R/W	See section	55.5.64/ 3160
400D_8FDC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL55_8)	32	R/W	See section	55.5.65/ 3161
400D_8FE0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL55_9)	32	R/W	See section	55.5.66/ 3161
400D_9000	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL56_1)	32	R/W	See section	55.5.58/ 3154
400D_9004	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL56_2)	32	R/W	See section	55.5.59/ 3155
400D_9008	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL56_3)	32	R/W	See section	55.5.60/ 3155
400D_900C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL56_4)	32	R/W	See section	55.5.61/ 3156
400D_9010	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL56_5)	32	R/W	See section	55.5.62/ 3158
400D_9014	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL56_6)	32	R/W	See section	55.5.63/ 3159
400D_9018	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL56_7)	32	R/W	See section	55.5.64/ 3160
400D_901C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL56_8)	32	R/W	See section	55.5.65/ 3161
400D_9020	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL56_9)	32	R/W	See section	55.5.66/ 3161
400D_9040	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL57_1)	32	R/W	See section	55.5.58/ 3154
400D_9044	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL57_2)	32	R/W	See section	55.5.59/ 3155
400D_9048	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL57_3)	32	R/W	See section	55.5.60/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_904C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL57_4)	32	R/W	See section	55.5.61/ 3156
400D_9050	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL57_5)	32	R/W	See section	55.5.62/ 3158
400D_9054	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL57_6)	32	R/W	See section	55.5.63/ 3159
400D_9058	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL57_7)	32	R/W	See section	55.5.64/ 3160
400D_905C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL57_8)	32	R/W	See section	55.5.65/ 3161
400D_9060	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL57_9)	32	R/W	See section	55.5.66/ 3161
400D_9080	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL58_1)	32	R/W	See section	55.5.58/ 3154
400D_9084	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL58_2)	32	R/W	See section	55.5.59/ 3155
400D_9088	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL58_3)	32	R/W	See section	55.5.60/ 3155
400D_908C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL58_4)	32	R/W	See section	55.5.61/ 3156
400D_9090	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL58_5)	32	R/W	See section	55.5.62/ 3158
400D_9094	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL58_6)	32	R/W	See section	55.5.63/ 3159
400D_9098	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL58_7)	32	R/W	See section	55.5.64/ 3160
400D_909C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL58_8)	32	R/W	See section	55.5.65/ 3161
400D_90A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL58_9)	32	R/W	See section	55.5.66/ 3161
400D_90C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL59_1)	32	R/W	See section	55.5.58/ 3154
400D_90C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL59_2)	32	R/W	See section	55.5.59/ 3155
400D_90C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL59_3)	32	R/W	See section	55.5.60/ 3155
400D_90CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL59_4)	32	R/W	See section	55.5.61/ 3156
400D_90D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL59_5)	32	R/W	See section	55.5.62/ 3158
400D_90D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL59_6)	32	R/W	See section	55.5.63/ 3159
400D_90D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL59_7)	32	R/W	See section	55.5.64/ 3160

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_90DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL59_8)	32	R/W	See section	55.5.65/ 3161
400D_90E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL59_9)	32	R/W	See section	55.5.66/ 3161
400D_9100	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL60_1)	32	R/W	See section	55.5.58/ 3154
400D_9104	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL60_2)	32	R/W	See section	55.5.59/ 3155
400D_9108	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL60_3)	32	R/W	See section	55.5.60/ 3155
400D_910C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL60_4)	32	R/W	See section	55.5.61/ 3156
400D_9110	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL60_5)	32	R/W	See section	55.5.62/ 3158
400D_9114	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL60_6)	32	R/W	See section	55.5.63/ 3159
400D_9118	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL60_7)	32	R/W	See section	55.5.64/ 3160
400D_911C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL60_8)	32	R/W	See section	55.5.65/ 3161
400D_9120	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL60_9)	32	R/W	See section	55.5.66/ 3161
400D_9140	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL61_1)	32	R/W	See section	55.5.58/ 3154
400D_9144	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL61_2)	32	R/W	See section	55.5.59/ 3155
400D_9148	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL61_3)	32	R/W	See section	55.5.60/ 3155
400D_914C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL61_4)	32	R/W	See section	55.5.61/ 3156
400D_9150	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL61_5)	32	R/W	See section	55.5.62/ 3158
400D_9154	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL61_6)	32	R/W	See section	55.5.63/ 3159
400D_9158	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL61_7)	32	R/W	See section	55.5.64/ 3160
400D_915C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL61_8)	32	R/W	See section	55.5.65/ 3161
400D_9160	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL61_9)	32	R/W	See section	55.5.66/ 3161
400D_9180	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL62_1)	32	R/W	See section	55.5.58/ 3154
400D_9184	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL62_2)	32	R/W	See section	55.5.59/ 3155

Table continues on the next page...

DCU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400D_9188	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL62_3)	32	R/W	See section	55.5.60/ 3155
400D_918C	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL62_4)	32	R/W	See section	55.5.61/ 3156
400D_9190	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL62_5)	32	R/W	See section	55.5.62/ 3158
400D_9194	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL62_6)	32	R/W	See section	55.5.63/ 3159
400D_9198	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL62_7)	32	R/W	See section	55.5.64/ 3160
400D_919C	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL62_8)	32	R/W	See section	55.5.65/ 3161
400D_91A0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL62_9)	32	R/W	See section	55.5.66/ 3161
400D_91C0	Control Descriptor Ln_0 Register (DCU1_CTRLDESCL63_1)	32	R/W	See section	55.5.58/ 3154
400D_91C4	Control Descriptor Ln_1 Register (DCU1_CTRLDESCL63_2)	32	R/W	See section	55.5.59/ 3155
400D_91C8	Control Descriptor Ln_2 Register (DCU1_CTRLDESCL63_3)	32	R/W	See section	55.5.60/ 3155
400D_91CC	Control Descriptor Ln_3 Register (DCU1_CTRLDESCL63_4)	32	R/W	See section	55.5.61/ 3156
400D_91D0	Control Descriptor Ln_4 Register (DCU1_CTRLDESCL63_5)	32	R/W	See section	55.5.62/ 3158
400D_91D4	Control Descriptor Ln_5 Register (DCU1_CTRLDESCL63_6)	32	R/W	See section	55.5.63/ 3159
400D_91D8	Control Descriptor Ln_6 Register (DCU1_CTRLDESCL63_7)	32	R/W	See section	55.5.64/ 3160
400D_91DC	Control Descriptor Ln_7 Register (DCU1_CTRLDESCL63_8)	32	R/W	See section	55.5.65/ 3161
400D_91E0	Control Descriptor Ln_8 Register (DCU1_CTRLDESCL63_9)	32	R/W	See section	55.5.66/ 3161

55.5.1 Control Descriptor Cursor 1 Register (DCUx_CTRLDESCCURSOR1)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 0h offset

Bit	31–27	26–16	15–11	10–0
R	0	HEIGHT	0	WIDTH
W				
Reset	0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0

DCUx_CTRLDESCCURSOR1 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 HEIGHT	Height of the cursor in pixels.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–0 WIDTH	Width of the cursor in pixels.

55.5.2 Control Descriptor Cursor 2 Register (DCUx_CTRLDESCCURSOR2)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 4h offset

Bit	31–27	26–16	15–11	10–0
R	0	POSY	0	POSX
W				
Reset	0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0

DCUx_CTRLDESCCURSOR2 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 POSY	Y position of the cursor in pixels
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–0 POSX	X position of the cursor in pixels

55.5.3 Control Descriptor Cursor 3 Register (DCUx_CTRLDESCCURSOR3)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CUR_EN	0							DEFAULT_CURSOR_COLOR[23:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEFAULT_CURSOR_COLOR[23:0]															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_CTRLDESCCURSOR3 field descriptions

Field	Description
31 CUR_EN	Cursor Enable signal. 0 Cursor is disabled 1 Enable the cursor
30–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DCUx_CTRLDESCCURSOR3 field descriptions (continued)

Field	Description
23–0 DEFAULT_ CURSOR_ COLOR[23:0]	Default pixel color value for the cursor. In the DCU4, the pixel value for the cursor is fixed for a particular frame.

55.5.4 Control Descriptor Cursor 4 Register (DCUx_CTRLDESCCURSOR4)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								HWC_BLINK_OFF							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							EN_BLINK	HWC_BLINK_ON							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_CTRLDESCCURSOR4 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 HWC_BLINK_ OFF	HWC blink register. Loads the counter value (number of frames) for which the cursor will remain turned OFF.
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 EN_BLINK	Enable the cursor blink mode. 0 Disable the blink mode 1 Enable the blink mode

Table continues on the next page...

DCUx_CTRLDESCCURSOR4 field descriptions (continued)

Field	Description
7–0 HWC_BLINK_ON	HWC blink register. Loads the counter value (number of frames) for which the cursor will remain turned ON.

55.5.5 DCU4 Mode Register (DCUx_DCU_MODE)

NOTE

Note: A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	DCU_SW_RESET	EN_DITHER	ADDB		ADDG		ADDR		DDR_MODE	BLEND_ITER			PDI_SYNC_LOCK			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI_INTERPOL_EN	RASTER_EN	PDI_EN	PDI_BYTE_REV	PDI_DE_MODE	PDI_NARROW_MODE	PDI_MODE		PDI_SLAVE_MODE	TAG_EN	SIG_EN	RESERVED	0	EN_GAMMA	DCU_MODE	
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_DCU_MODE field descriptions

Field	Description
31 DCU_SW_RESET	Puts the DCU4 internal configuration into its reset state. This has the effect of stopping all display activity except the PCLK; graphics are not fetched and sent to the panel. 0 No action 1 All DCU4 internal registers are forced into their reset state. User registers are not affected
30 EN_DITHER	Enable dithering mode. 0 Disabled 1 Enabled
29–28 ADDB	Two-bit value to be added to pixel blue component for dithering.

Table continues on the next page...

DCUx_DCU_MODE field descriptions (continued)

Field	Description
27–26 ADDG	Two bit Value to be added with pixel green component for dithering.
25–24 ADDR	Two-bit value to be added to pixel red component for dithering.
23 DDR_MODE	Enables Special DDR Mode (see Section 54.6.9 Special DDR Mode"). Select BLEND_ITER = 6 when enabled. 0 Off 1 On
22–20 BLEND_ITER	Defines the maximum number of pixels which are blended in the pixel blend stack. 000 Reserved 001 Reserved 010 Two pixel blending (default) 011 Three pixel blending 100 Four pixel blending 101 Five pixel blending 110 Six plane blending 111 Reserved
19–16 PDI_SYNC_ LOCK	Defines the number of frames which should be received by the PDI validation state machine before it locks and sets the PDI_LOCK_DET bit in the PDI Status Register (see PDI Status Register (DCU_PDI_STATUS))
15 PDI_INTERPOL_ EN	Control Bit to decide whether the conversion from YCbCr 4:2:2 to 4:4:4 needs to be done using interpolation or Chroma value is same for two pixels. 0 Chroma value is same for two pixels 1 Interpolation is enabled
14 RASTER_EN	Enables raster scanning of pixel data including the VSYNC and HSYNC signals and the pixel data. This bit takes effect immediately and does not require a transfer to the frame timing logic. 0 Disabled 1 Enabled
13 PDI_EN	Enables the PDI. 0 Disabled 1 Enabled
12 PDI_BYTE_REV	Controls the byte ordering in Narrow Mode. 0 LSB is followed by MSB data 1 MSB is followed by LSB data
11 PDI_DE_MODE	Enables the PDI data Enable Mode. Here Data Enable is treated as an input. 0 Value on data Enable signal is ignored 1 Data enable controls the write to the PDI FIFO
10 PDI_NARROW_ MODE	Enables the PDI Narrow Mode. Refer to "Normal and narrow mode" section.

Table continues on the next page...

DCUx_DCU_MODE field descriptions (continued)

Field	Description
	0 Narrow Mode is Disabled 1 Narrow Mode is Enabled
9–8 PDI_MODE	Defines the different modes in which PDI is operating. 00 8-bit monochrome data input 01 16-bit RGB 565 format 10 18-bit RGB 666 data format 11 YCbCr data in 4:2:2 format
7 PDI_SLAVE_ MODE	Enables PDI slave Mode. 0 Disabled 1 Enabled
6 TAG_EN	Enables the calculation of CRC only on the safety layers. 0 CRC calculated over the whole area of interest (area of interest given by SIG_DESC registers) 1 Calculates CRC only on safety enabled layers
5 SIG_EN	Enables the signature calculator block. 0 Signature calculator is disabled 1 Signature calculator is enabled
4 RESERVED	This field is reserved. 0 External Synchronization. The PDI receives the SYNC (HSYNC, VSYNC) signals from external source. 1 Internal Synchronization. PDI extracts the SYNC information from the digital RGB data. **NOTE:** YCbCr Mode supports Internal Sync only. Therefore, when PDI_ MODE = 3, PDI_SYNC must be set to 0.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 EN_GAMMA	Enables/Disables the Gamma Correction. 0 Gamma correction is disabled 1 Gamma Correction is enabled
1–0 DCU_MODE	DCU operating mode. 00 DCU off (pixel clock active if enabled by I/O). 01 Normal mode. Panel content controlled by layer configuration. 10 Test mode. DCU disables all DMA fetches and all the pixels of an enabled layer take the value in the CLUT RAM selected by the respective LUOFFS field of control descriptor 4. 11 Color Bar Generation. Panel content controlled by color bar registers.

55.5.6 Background Register (DCUx_BGND)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								BGND_R								BGND_G								BGND_B							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_BGND field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 BGND_R	Red component of the default color displayed in the sectors where no layer is active.
15–8 BGND_G	Green component of the default color displayed in the sectors where no layer is active.
7–0 BGND_B	Blue component of the default color displayed in the sectors where no layer is active.

55.5.7 Display Size Register (DCUx_DISP_SIZE)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

NOTE

The device supports the panel size up to XGA (1024 x 768) only.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					DELTA_Y											0									DELTA_X						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_DISP_SIZE field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 DELTA_Y	Sets the display size vertical resolution (in pixels).
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 DELTA_X	Sets the display size horizontal resolution (in multiples of 16 pixels).

55.5.8 Horizontal Sync Parameter Register (DCUx_HSYN_PARA)

HSYN_PARA register sets timing parameters related to the horizontal synchronization signal generation. The fields FP_H, BP_H, and PW_H stand for HSYNC signal front-porch, back-porch, and active pulse width, respectively.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	BP_H									0		PW_H			
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PW_H					0		FP_H								
W																
Reset	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1

DCUx_HSYN_PARA field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–22 BP_H	HSYNC back-porch pulse width (in pixel clock cycles). Pulse width has a minimum value of 1.
21–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–11 PW_H	HSYNC active pulse width (in pixel clock cycles).
10–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DCUx_HSYN_PARA field descriptions (continued)

Field	Description
8–0 FP_H	HSYNC front-porch pulse width (in pixel clock cycles). Pulse width has a minimum value of 1.

55.5.9 Vertical Sync Parameter Register (DCUx_VSYN_PARA)

VSYN_PARA register sets timing parameters related to the vertical synchronization signal generation. The fields FP_V, BP_V, and PW_V stand for VSYNC signal front-porch, back-porch, and active pulse width, respectively.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	BP_V									0		PW_V			
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PW_V					0		FP_V								
W																
Reset	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1

DCUx_VSYN_PARA field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–22 BP_V	VSYNC back-porch pulse width (in horizontal line cycles). Pulse width has a minimum value of 1.
21–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–11 PW_V	VSYNC active pulse width (in horizontal line cycles).
10–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8–0 FP_V	VSYNC front-porch pulse width (in horizontal line cycles). Pulse width has a minimum value of 1.

55.5.10 Synchronize Polarity Register (DCUx_SYNPOL)

The SYNPOL register selects the polarity for the DCU synchronisation signals (HSYNC and VSYNC).

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					INV_PDI_DE	INV_PDI_HS	INV_PDI_VS	INV_PDI_CLK	INV_PXCK	NEG	RESERVED	RESERVED	RESERVED	INV_VS	INV_HS
W												RESERVED	RESERVED	RESERVED		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_SYNPOL field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 INV_PDI_DE	Polarity of PDI input data Enable. 0 DE is active low 1 DE is active high

Table continues on the next page...

DCUx_SYNPOL field descriptions (continued)

Field	Description
9 INV_PDI_HS	Polarity of PDI input HSYNC. 0 HSYNC is active low 1 HSYNC is active high
8 INV_PDI_VS	Polarity of PDI input VSYNC. 0 VSYNC is active low 1 VSYNC is active high
7 INV_PDI_CLK	Polarity of PDI input Clock. 0 DCU4 samples data on the rising edge 1 DCU4 samples data on the falling edge
6 INV_PXCK	Polarity change of Pixel Clock. 0 Display samples data on the falling edge 1 Display samples data on the rising edge
5 NEG	Indicates if value at the output (pixel data output) needs to be negated. 0 Output is to remain same 1 Output to be negated
4 RESERVED	This field is reserved.
3 RESERVED	This field is reserved.
2 RESERVED	This field is reserved.
1 INV_VS	Invert Vertical synchronization signal. 0 VSYNC signal not inverted (active HIGH). 1 Invert VSYNC signal (active LOW).
0 INV_HS	Invert Horizontal synchronization signal. 0 HSYNC signal not inverted (active HIGH). 1 Invert HSYNC signal (active LOW).

55.5.11 Threshold Register (DCUx_THRESHOLD)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					LS_BF_VS										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUT_BUF_HIGH								OUT_BUF_LOW							
W																
Reset	0	1	1	1	1	0	0	0	0	0	0	0	1	0	1	0

DCUx_THRESHOLD field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 LS_BF_VS	Lines before VS_BLANK threshold value. The LS_BF_VS status flag (in INT_STATUS) is set this number of lines before the VS_BLANK signal is asserted.
15–8 OUT_BUF_HIGH	Output buffer high threshold (in pixels). When the output buffer exceeds this value the datapath clock is suspended.
7–0 OUT_BUF_LOW	Output buffer filling low Threshold (in pixels).This value is used to generate the underrun exception (UNDRUN in INT_STATUS).

55.5.12 Interrupt Status Register (DCUx_INT_STATUS)

See Interrupt generation for a description of how the DCU4 collects interrupt events into different source groups.

Unless stated otherwise, the flags in this register are cleared by writing 1 to the relevant bit.

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	P6_EMPTY	P5_EMPTY	P4_EMPTY	P3_EMPTY	P2_EMPTY	P1_EMPTY	0		P6_FIFO_HI_FLAG	P6_FIFO_LO_FLAG	P5_FIFO_HI_FLAG	P5_FIFO_LO_FLAG	P4_FIFO_HI_FLAG	P4_FIFO_LO_FLAG	P3_FIFO_HI_FLAG	P3_FIFO_LO_FLAG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	DMA_TRANS_FINISH	0	LYR_TRANS_FINISH	IPM_ERROR	PROG_END	P2_FIFO_HI_FLAG	P2_FIFO_LO_FLAG	P1_FIFO_HI_FLAG	P1_FIFO_LO_FLAG	CRC_OVERFLOW	CRC_READY	VS_BLANK	LS_BF_VS	UNDRUN	VSYNC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_INT_STATUS field descriptions

Field	Description
31 P6_EMPTY	Interrupt flag to indicate that the FIFO in position 6 in the pixel blend stack underflowed.
30 P5_EMPTY	Interrupt flag to indicate that the FIFO in position 5 in the pixel blend stack underflowed.
29 P4_EMPTY	Interrupt flag to indicate that the FIFO in position 4 in the pixel blend stack underflowed.
28 P3_EMPTY	Interrupt flag to indicate that the FIFO in position 3 in the pixel blend stack underflowed.
27 P2_EMPTY	Interrupt flag to indicate that the FIFO in position 2 in the pixel blend stack underflowed.
26 P1_EMPTY	Interrupt flag to indicate that the FIFO in position 1 (lowest) in the pixel blend stack underflowed.
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 P6_FIFO_HI_FLAG	Interrupt flag to indicate that the high threshold has been reached in the FIFO in position 6 in the pixel blend stack.
22 P6_FIFO_LO_FLAG	Interrupt flag to indicate that the low threshold has been reached in the FIFO in position 6 in the pixel blend stack.
21 P5_FIFO_HI_FLAG	Interrupt flag to indicate that the high threshold has been reached in the FIFO in position 5 in the pixel blend stack.
20 P5_FIFO_LO_FLAG	Interrupt flag to indicate that the low threshold has been reached in the FIFO in position 5 in the pixel blend stack.
19 P4_FIFO_HI_FLAG	Interrupt flag to indicate that the high threshold has been reached in the FIFO in position 4 in the pixel blend stack.
18 P4_FIFO_LO_FLAG	Interrupt flag to indicate that the low threshold has been reached in the FIFO in position 4 in the pixel blend stack.
17 P3_FIFO_HI_FLAG	Interrupt flag to indicate that the high threshold has been reached in the FIFO in position 3 in the pixel blend stack.
16 P3_FIFO_LO_FLAG	Interrupt flag to indicate that the low threshold has been reached in the FIFO in position 3 in the pixel blend stack.

Table continues on the next page...

DCUx_INT_STATUS field descriptions (continued)

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 DMA_TRANS_ FINISH	Interrupt flag, which indicates that the DCU4 DMA has fetched the last pixel of data from the memory.
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 LYR_TRANS_ FINISH	Interrupt flag to indicate that the transfer is complete of layer configuration from the layer control descriptor registers into the DCU4 functional block
11 IPM_ERROR	Interrupt flag, which indicates that an error has occured in the Magenta line transaction.
10 PROG_END	Interrupt flag which indicates that the DCU4 has begun to transfer layer configuration from the layer control descriptor registers into the DCU4 functional block. Any register modification after this time and before LYR_TRANS_FINISH is asserted may or may not be included in this transfer.
9 P2_FIFO_HI_ FLAG	Interrupt flag to indicate that the high threshold has been reached in the FIFO in position 2 in the pixel blend stack.
8 P2_FIFO_LO_ FLAG	Interrupt flag to indicate that the low threshold has been reached in the FIFO in position 2 in the pixel blend stack.
7 P1_FIFO_HI_ FLAG	Interrupt flag to indicate that the high threshold has been reached in the FIFO in position 1 (lowest) in the pixel blend stack.
6 P1_FIFO_LO_ FLAG	Interrupt flag to indicate that the low threshold has been reached in the FIFO in position 1 (lowest) in the pixel blend stack.
5 CRC_ OVERFLOW	Interrupt signal to indicate that CRC_ready has not been serviced and CRC has been calculated for the next frame
4 CRC_READY	Interrupt flag to indicate CRC calculation is done and ready to be compared with precomputed CRC value by the software.
3 VS_BLANK	Interrupt signal to indicate vertical blanking period. This is the period in which all the registers that affect the visible state of the layers need to be latched. This is needed so that CPU writes to the register while the display is being updated does not cause any errors. Interrupt can be cleared by writing 1 to this bit..
2 LS_BF_VS	Interrupt flag to indicate the Lines Before VS_BLANK event has been reached. The LS_BF_VS field in the Threshold register defines the timing of the event.
1 UNDRUN	Interrupt flag to indicate the output buffer underrun condition. Asserted when the panel needs data and the output buffer level is lower than or equal to the OUT_BUF_LOW threshold. Flag is cleared when the data in the output buffer is greater than threshold and CPU writes 1 to this bit.
0 VSYNC	Interrupt flag to indicate that the vertical synchronization phase has begun. If enabled, an interrupt is generated at the beginning of a frame.

55.5.13 Interrupt Mask Register (DCUx_INT_MASK)

This register enables or masks the corresponding interrupt.

For each individual interrupt bit:

0 Not masked

1 Interrupt is masked

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M_P6_EMPTY	M_P5_EMPTY	M_P4_EMPTY	M_P3_EMPTY	M_P2_EMPTY	M_P1_EMPTY	0		M_P6_FIFO_HI_FLAG	M_P6_FIFO_LO_FLAG	M_P5_FIFO_HI_FLAG	M_P5_FIFO_LO_FLAG	M_P4_FIFO_HI_FLAG	M_P4_FIFO_LO_FLAG	M_P3_FIFO_HI_FLAG	M_P3_FIFO_LO_FLAG
W																
Reset	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	M_DMA_TRANS_FINISH	0	M_LYR_TRANS_FINISH	M_IPM_ERROR	M_PROG_END	M_P2_FIFO_HI_FLAG	M_P2_FIFO_LO_FLAG	M_P1_FIFO_HI_FLAG	M_P1_FIFO_LO_FLAG	M_CRC_OVERFLOW	M_CRC_READY	M_VS_BLANK	M_LS_BF_VS	M_UNDRUN	M_VSYNC
W																
Reset	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1

DCUx_INT_MASK field descriptions

Field	Description
31 M_P6_EMPTY	Mask for P6_EMPTY interrupt flag.
30 M_P5_EMPTY	Mask for P5_EMPTY interrupt flag.
29 M_P4_EMPTY	Mask for P4_EMPTY interrupt flag.
28 M_P3_EMPTY	Mask for P3_EMPTY interrupt flag.
27 M_P2_EMPTY	Mask for P2_EMPTY interrupt flag.
26 M_P1_EMPTY	Mask for P1_EMPTY interrupt flag.
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 M_P6_FIFO_HI_FLAG	Mask for P6_FIFO_HI_FLAG interrupt flag.
22 M_P6_FIFO_LO_FLAG	Mask for P6_FIFO_LO_FLAG interrupt flag.

Table continues on the next page...

DCUx_INT_MASK field descriptions (continued)

Field	Description
21 M_P5_FIFO_HI_ FLAG	Mask for P5_FIFO_HI_FLAG interrupt flag.
20 M_P5_FIFO_LO_ FLAG	Mask for P5_FIFO_LO_FLAG interrupt flag.
19 M_P4_FIFO_HI_ FLAG	Mask for P4_FIFO_HI_FLAG interrupt flag.
18 M_P4_FIFO_LO_ FLAG	Mask for P4_FIFO_LO_FLAG interrupt flag.
17 M_P3_FIFO_HI_ FLAG	Mask for P3_FIFO_HI_FLAG interrupt flag.
16 M_P3_FIFO_LO_ FLAG	Mask for P6_FIFO_LO_FLAG interrupt flag.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 M_DMA_ TRANS_FINISH	Mask for DMA_TRANS_FINISH interrupt flag.
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 M_LYR_TRANS_ FINISH	Mask for SHDW_CMPLT interrupt flag.
11 M_IPM_ERROR	Mask for IPM_ERROR interrupt flag.
10 M_PROG_END	Mask for PROG_END interrupt flag.
9 M_P2_FIFO_HI_ FLAG	Mask for P2_FIFO_HI_FLAG interrupt flag.
8 M_P2_FIFO_LO_ FLAG	Mask for P2_FIFO_LO_FLAG interrupt flag.
7 M_P1_FIFO_HI_ FLAG	Mask for P1_FIFO_HI_FLAG interrupt flag.
6 M_P1_FIFO_LO_ FLAG	Mask for P1_FIFO_LO_FLAG interrupt flag.
5 M_CRC_ OVERFLOW	Mask for CRC_OVERFLOW interrupt flag.

Table continues on the next page...

DCUx_INT_MASK field descriptions (continued)

Field	Description
4 M_CRC_READY	Mask for CRC_READY interrupt flag.
3 M_VS_BLANK	Mask for VS_BLANK interrupt flag.rupt can be cleared by writing 1 to this bit..
2 M_LS_BF_VS	Mask for LS_BF_VS interrupt flag.
1 M_UNDRUN	Mask for M_UNDRUN interrupt flag.
0 M_VSYNC	Mask for VSYNC interrupt flag.

55.5.14 COLBAR_1 Register (DCUx_COLBAR_1)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 34h offset

Bit	31–24	23–16	15–8	7–0
R	1	COLBAR_1_R	COLBAR_1_G	COLBAR_1_B
W				
Reset	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

DCUx_COLBAR_1 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
23–16 COLBAR_1_R	Red component value.
15–8 COLBAR_1_G	Green component value.
7–0 COLBAR_1_B	Blue component value.

55.5.15 COLBAR_2 Register (DCUx_COLBAR_2)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1								COLBAR_2_R								COLBAR_2_G								COLBAR_2_B							
W																																
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

DCUx_COLBAR_2 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
23–16 COLBAR_2_R	Red component value.
15–8 COLBAR_2_G	Green component value.
7–0 COLBAR_2_B	Blue component value.

55.5.16 COLBAR_3 Register (DCUx_COLBAR_3)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 3Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1								COLBAR_3_R								COLBAR_3_G								COLBAR_3_B							
W																																
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

DCUx_COLBAR_3 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
23–16 COLBAR_3_R	Red component value.
15–8 COLBAR_3_G	Green component value.
7–0 COLBAR_3_B	Blue component value.

55.5.17 COLBAR_4 Register (DCUx_COLBAR_4)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1								COLBAR_4_R								COLBAR_4_G								COLBAR_4_B							
W																																
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

DCUx_COLBAR_4 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
23–16 COLBAR_4_R	Red component value.
15–8 COLBAR_4_G	Green component value.
7–0 COLBAR_4_B	Blue component value.

55.5.18 COLBAR_5 Register (DCUx_COLBAR_5)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1								COLBAR_5_R								COLBAR_5_G								COLBAR_5_B							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

DCUx_COLBAR_5 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
23–16 COLBAR_5_R	Red component value.
15–8 COLBAR_5_G	Green component value.
7–0 COLBAR_5_B	Blue component value.

55.5.19 COLBAR_6 Register (DCUx_COLBAR_6)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 48h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1								COLBAR_6_R								COLBAR_6_G								COLBAR_6_B							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_COLBAR_6 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
23–16 COLBAR_6_R	Red component value.
15–8 COLBAR_6_G	Green component value.
7–0 COLBAR_6_B	Blue component value.

55.5.20 COLBAR_7 Register (DCUx_COLBAR_7)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 4Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1								COLBAR_7_R								COLBAR_7_G								COLBAR_7_B							
W																																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

DCUx_COLBAR_7 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
23–16 COLBAR_7_R	Red component value.
15–8 COLBAR_7_G	Green component value.
7–0 COLBAR_7_B	Blue component value.

55.5.21 COLBAR_8 Register (DCUx_COLBAR_8)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 50h offset

Bit	31–24	23–16	15–8	7–0
R	1	COLBAR_8_R	COLBAR_8_G	COLBAR_8_B
W				
Reset	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1

DCUx_COLBAR_8 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
23–16 COLBAR_8_R	Red component value.
15–8 COLBAR_8_G	Green component value.
7–0 COLBAR_8_B	Blue component value.

55.5.22 Divide Ratio Register (DCUx_DIV_RATIO)

This register for vertical/horizontal size of the area for CRC calculation.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 54h offset

Bit	31–8	7–0
R	0	DIV_RATIO
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

DCUx_DIV_RATIO field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 DIV_RATIO	Specifies the divide value for the input clock. Used to generate the pixel clock to support different types of displays. To divide by N, set the DIV_RATIO to (N – 1).

55.5.23 Sign Calculation 1 Register (DCUx_SIGN_CALC_1)

This register specifies the vertical/horizontal size of the area for the CRC calculation.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 58h offset

Bit	31–27	26–16	15–11	10–0
R	0	SIG_VER_SIZE	0	SIG_HOR_SIZE
W				
Reset	0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0

DCUx_SIGN_CALC_1 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 SIG_VER_SIZE	Vertical size of the window of interest of pixels for CRC calculation (in pixels).
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–0 SIG_HOR_SIZE	Horizontal size of window of interest of pixels for CRC calculations (in pixels).

55.5.24 Sign Calculation 2 Register (DCUx_SIGN_CALC_2)

This register specifies the position of the window of interest for the CRC calculation

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 5Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					SIG_VER_POS											0					SIG_HOR_POS										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_SIGN_CALC_2 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 SIG_VER_POS	Vertical position of the window of interest of pixels for CRC calculation (in pixels).
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–0 SIG_HOR_POS	Horizontal position of window of interest of pixels for CRC calculation (in pixels).

55.5.25 CRC Value Register (DCUx_CRC_VAL)

This register contains the result of the CRC calculation for the value of the pixels on the safety layers.

Address: Base address + 60h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC_VAL																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_CRC_VAL field descriptions

Field	Description
31–0 CRC_VAL	The result of the CRC calculation for the value of the pixels on the safety layers

55.5.26 PDI Status Register (DCUx_PDI_STATUS)

Address: Base address + 64h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						PDI_BLANKING_ERR	PDI_ECC_ERR2	PDI_ECC_ERR1	PDI_LOCK_LOST	PDI_LOCK_DET	PDI_VSYNC_DET	PDI_HSYNC_DET	PDI_DE_DET	PDI_CLK_LOST	PDI_CLK_DET
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_PDI_STATUS field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 PDI_BLANKING_ERR	Status bit to inform the software that 80h,10h sequence is not present during the blanking period in internal sync mode. 0 Correct data sequence present in blanking period. 1 Correct data sequence not present in blanking period.
8 PDI_ECC_ERR2	Status bit to inform the software about multibit bit error that is detected. 0 Multibit ECC error is not detected 1 Multibit ECC error detected
7 PDI_ECC_ERR1	Status bit to inform the software about one bit error is detected. 0 One bit ECC error is not detected. 1 One bit ECC error detected
6 PDI_LOCK_LOST	Status bit to inform the software that frame lock is lost. 0 Frame is locked 1 Frame lock is lost
5 PDI_LOCK_DET	Status bit to inform the software PDI is frame locked to the camera interface. 0 Waiting for frame to lock 1 Frame lock is detected
4 PDI_VSYNC_DET	Status bit to inform the software that vsync for the camera data has been detected.

Table continues on the next page...

DCUx_PDI_STATUS field descriptions (continued)

Field	Description
	0 pdi_vsync not detected 1 pdi_vsync is detected
3 PDI_HSYNC_ DET	Status bit to inform the software that hsync for the camera data has been detected. 0 pdi_hsync not detected 1 pdi_hsync is detected
2 PDI_DE_DET	Status bit to inform the software that data Enable for the camera data has been detected. 0 pdi_de not detected 1 pdi_de is detected
1 PDI_CLK_LOST	Status bit to inform the software that pdi_clk is lost. 0 pdi_clk is present 1 pdi_clk is lost
0 PDI_CLK_DET	Status bit to inform the software that clock for the camera data has been detected. 0 pdi_clk not detected 1 pdi_clk is detected

55.5.27 PDI Status Mask Register (DCUx_PDI_STA_MSK)

This register enables or masks the corresponding interrupt.

For each individual interrupt bit:

0 Not masked

1 Interrupt is masked

Address: Base address + 68h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						M_PDI_ BLANKING_ERR	M_PDI_ECC_ERR2	M_PDI_ECC_ERR1	M_PDI_LOCK_ LOST	M_PDI_LOCK_DET	M_PDI_VSYNC_ DET	M_PDI_HSYNC_ DET	M_PDI_DE_DET	M_PDI_CLK_LOST	M_PDI_CLK_DET
W																
Reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

DCUx_PDI_STA_MSK field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 M_PDI_ BLANKING_ERR	Mask for PDI_BLANKING_ERR interrupt flag.
8 M_PDI_ECC_ ERR2	Mask for PDI_ECC_ERR2 interrupt flag.
7 M_PDI_ECC_ ERR1	Mask for PDI_ECC_ERR1 interrupt flag.
6 M_PDI_LOCK_ LOST	Mask for PDI_LOCK_LOST interrupt flag.
5 M_PDI_LOCK_ DET	Mask for PDI_LOCK_DET interrupt flag.
4 M_PDI_VSYNC_ DET	Mask for PDI_VSYNC_DET interrupt flag.
3 M_PDI_HSYNC_ DET	Mask for PDI_HSYNC_DET interrupt flag.
2 M_PDI_DE_DET	Mask for PDI_DE_DET interrupt flag.
1 M_PDI_CLK_ LOST	Mask for PDI_CLK_LOST interrupt flag.
0 M_PDI_CLK_ DET	Mask for PDI_CLK_DET interrupt flag.

55.5.28 Parameter Error Status 1 Register (DCUx_PARR_ERR_STATUS1)

An error in a layer can occur under the following conditions:

- Number of pixels in a tile > maximum tile memory size when internal tile memory mode is in use.
- There is a non-valid horizontal size and color format combination. See Layer size and positioning for details.

These errors are grouped into a single bit error for each layer. The parameter error specific to each layer is signalled only when the layer is enabled.

Address: Base address + 6Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	L[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_PARR_ERR_STATUS1 field descriptions

Field	Description
31–0 L[31:0]	For each individual interrupt: 0 Parameter error is not set 1 Parameter error is set

55.5.29 Parameter Error Status 2 Register (DCUx_PARR_ERR_STATUS2)

An error in a layer can occur under the following conditions:

- Number of pixels in a tile > maximum tile memory size when internal tile memory mode is in use.
- There is a non-valid horizontal size and color format combination. See Layer size and positioning for details.

These errors are grouped into a single bit error for each layer. The parameter error specific to each layer is signalled only when the layer is enabled.

For each individual interrupt:

0 Parameter error is not set

1 Parameter error is set

Address: Base address + 70h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	L[63:32]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_PARR_ERR_STATUS2 field descriptions

Field	Description
31–0 L[63:32]	For each individual interrupt: 0 Parameter error is not set 1 Parameter error is set

55.5.30 Parameter Error Status 3 Register (DCUx_PARR_ERR_STATUS3)

DISP_ERR occurs when the size of display (height or width) is set to zero or when the pulse width of hsync/vsync is programmed as zero.

SIG_ERR occurs when the area of interest for calculating CRC value is programmed with values which are outside the display.

HWC_ERR occurs if size of cursor programmed is greater than memory size (256 x 32). See Hardware cursor for further details on how cursor can be programmed.

RLE_ERR occurs when more than one layer has its RLE_EN bit set (Control Descriptor 4).

Address: Base address + 7Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												RLE_ERR	HWC_ERR	SIG_ERR	DISP_ERR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_PARR_ERR_STATUS3 field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 RLE_ERR	Error signal to indicate that more than one layer has RLE mode enabled.
2 HWC_ERR	Interrupt signal to indicate HWC error. This can occur if HWC position is out of display area or cursor memory is bigger than the HWC size. When this occurs, the HWC is disabled.

Table continues on the next page...

DCUx_PARR_ERR_STATUS3 field descriptions (continued)

Field	Description
1 SIG_ERR	Interrupt occurs whenever the area of interest specified by SIG_CALC register is outside the display size.
0 DISP_ERR	Interrupt occurs whenever width and height of display, pulse width (both vertical and horizontal sync) value is 0.

55.5.31 Mask Parameter Error Status 1 Register (DCUx_MASK_PARR_ERR_STATUS1)

This register enables or masks the corresponding interrupt.

For each individual interrupt bit:

0 Not masked

1 Interrupt is masked

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	M_L_PARR_ERR[31:0]																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

DCUx_MASK_PARR_ERR_STATUS1 field descriptions

Field	Description
31–0 M_L_PARR_ ERR[31:0]	Mask for L[31:0] interrupt flag.

55.5.32 Mask Parameter Error Status 2 Register (DCUx_MASK_PARR_ERR_STATUS2)

This register enables or masks the corresponding interrupt.

For each individual interrupt bit:

0 Not masked

1 Interrupt is masked

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	M_L_PARR_ERR[63:32]																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

DCUx_MASK_PARR_ERR_STATUS2 field descriptions

Field	Description
31–0 M_L_PARR_ ERR[63:32]	Mask for L[63:32] interrupt flag.

55.5.33 Mask Parameter Error Status 3 Register (DCUx_MASK_PARR_ERR_STATUS3)

This register enables or masks the corresponding interrupt.

For each individual interrupt bit:

0 Not masked

1 Interrupt is masked

Address: Base address + 90h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												M_RLE_ERR	M_HWC_ERR	M_SIG_ERR	M_DISP_ERR
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

DCUx_MASK_PARR_ERR_STATUS3 field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 M_RLE_ERR	Mask for RLE_ERR interrupt flag.
2 M_HWC_ERR	Mask for HWC_ERR interrupt flag.
1 M_SIG_ERR	Mask for SIG_ERR interrupt flag.
0 M_DISP_ERR	Mask for DISP_ERR interrupt flag.

55.5.34 Threshold Input 1 Register (DCUx_THRESHOLD_INP_BUF_1)

This is the threshold configuration register for FIFO buffers 1 and 2. Note: A write to this register does not take effect until the value is transferred to the frame timing logic - see 43.5.4.2 Transfer of DCU Configuration.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 94h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	INP_BUF_P2_HI							0	INP_BUF_P2_LO						
W																
Reset	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	INP_BUF_P1_HI							Reserved	INP_BUF_P1_LO						
W																
Reset	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

DCUx_THRESHOLD_INP_BUF_1 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–24 INP_BUF_P2_HI	High threshold for the FIFO in position 2 in the pixel blend stack.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–16 INP_BUF_P2_LO	Low threshold for the FIFO in position 2 in the pixel blend stack.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 INP_BUF_P1_HI	High threshold for the FIFO in position 1 (lowest) in the pixel blend stack.
7 Reserved	This field is reserved.
6–0 INP_BUF_P1_LO	Low threshold for the FIFO in position 1 (lowest) in the pixel blend stack.

55.5.35 Threshold Input 2 Register (DCUx_THRESHOLD_INP_BUF_2)

This is the threshold configuration register for FIFO buffers 3 and 4.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 98h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	INP_BUF_P4_HI							0	INP_BUF_P4_LO						
W																
Reset	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	INP_BUF_P3_HI							Reserved	INP_BUF_P3_LO						
W																
Reset	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

DCUx_THRESHOLD_INP_BUF_2 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–24 INP_BUF_P4_HI	High threshold for the FIFO in position 4 in the pixel blend stack.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–16 INP_BUF_P4_LO	Low threshold for the FIFO in position 4 in the pixel blend stack.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 INP_BUF_P3_HI	High threshold for the FIFO in position 3 in the pixel blend stack.
7 Reserved	This field is reserved.
6–0 INP_BUF_P3_LO	Low threshold for the FIFO in position 3 in the pixel blend stack.

55.5.36 Threshold Input 3 Register (DCUx_THRESHOLD_INP_BUF_3)

This is the threshold configuration register for FIFO buffers 5 and 6.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 9Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	INP_BUF_P6_HI							0	INP_BUF_P6_LO						
W																
Reset	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	INP_BUF_P5_HI							Reserved	INP_BUF_P5_LO						
W																
Reset	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

DCUx_THRESHOLD_INP_BUF_3 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–24 INP_BUF_P6_HI	High threshold for the FIFO in position 6 in the pixel blend stack.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–16 INP_BUF_P6_LO	Low threshold for the FIFO in position 6 in the pixel blend stack.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 INP_BUF_P5_HI	High threshold for the FIFO in position 5 in the pixel blend stack.
7 Reserved	This field is reserved.
6–0 INP_BUF_P5_LO	Low threshold for the FIFO in position 5 in the pixel blend stack.

55.5.37 LUMA Component Register (DCUx_LUMA_COMP)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + A0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Y_RED										0	Y_GREEN				
W																
Reset	1	0	0	1	0	1	0	1	0	0	0	1	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Y_GREEN					0	Y_BLUE									
W																
Reset	1	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0

DCUx_LUMA_COMP field descriptions

Field	Description
31–22 Y_RED	Luminance coefficient for red component.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–11 Y_GREEN	Luminance coefficient for green component.
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–0 Y_BLUE	Luminance coefficient for blue component.

55.5.38 Red Chroma Components Register (DCUx_CHROMA_RED)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + A4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					CR_RED											0				CB_RED											
W																																
Reset	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_CHROMA_RED field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 CR_RED	Cr coefficient for calculation of red component.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 CB_RED	Cb coefficient for calculation of red component.

55.5.39 Green Chroma Components Register (DCUx_CHROMA_GREEN)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + A8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					CR_GREEN											0				CB_GREEN											
W																																
Reset	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	0

DCUx_CHROMA_GREEN field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 CR_GREEN	Cr coefficient for calculation of green component.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 CB_GREEN	Cb coefficient for calculation of green component.

55.5.40 Blue Chroma Components Register (DCUx_CHROMA_BLUE)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + ACh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					CR_BLUE											0				CB_BLUE											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

DCUx_CHROMA_BLUE field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 CR_BLUE	Cr coefficient for calculation of blue component.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 CB_BLUE	Cb coefficient for calculation of blue component.

55.5.41 CRC Position Register (DCUx_CRC_POS)

This register contains the result of the CRC calculation for the position of the pixels on the safety layers.

Address: Base address + B0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC_POS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_CRC_POS field descriptions

Field	Description
31–0 CRC_POS	The result of the CRC calculation for the position of the pixels on the safety layers.

55.5.42 Layer Interpolation Enable Register (DCUx_LYR_INTPOL_EN)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + B4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EN	0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

DCUx_LYR_INTPOL_EN field descriptions

Field	Description
31 EN	Interpolation Enable bit for DCU3 Layer coded in YCbCr422 format. This bit controls whether the chroma value for eac h pixel in the conversion from YCbCr 4:2:2 to 4:4:4 should use interpolation or the same value for both pixels.

Table continues on the next page...

DCUx_LYR_INTPOL_EN field descriptions (continued)

Field	Description
	0 Chroma value is same for two pixels 1 Interpolation is enabled
30–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.43 Layer Luminance Component Register (DCUx_LYR_LUMA_COMP)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + B8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Y_RED										0	Y_GREEN				
W																
Reset	1	0	0	1	0	1	0	1	0	0	0	1	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Y_GREEN					0	Y_BLUE									
W																
Reset	1	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0

DCUx_LYR_LUMA_COMP field descriptions

Field	Description
31–22 Y_RED	Luminance coefficient for red component.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–11 Y_GREEN	Luminance coefficient for green component.
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–0 Y_BLUE	Luminance coefficient for blue component.

55.5.44 Layer Chroma Red Register (DCUx_LYR_CHRM_RED)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + BCh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					Cr_RED											0				Cb_RED											
W																																
Reset	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_LYR_CHRM_RED field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 Cr_RED	Cr coefficient for calculation of red component.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 Cb_RED	Cb coefficient for calculation of red component.

55.5.45 Layer Chroma Green Register (DCUx_LYR_CHRM_GRN)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + C0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					Cr_GREEN											0				Cb_GREEN											
W																																
Reset	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	0

DCUx_LYR_CHRM_GRN field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 Cr_GREEN	Cr coefficient for calculation of green component.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 Cb_GREEN	Cr coefficient for calculation of green component.

55.5.46 Layer Chroma Blue Register (DCUx_LYR_CHRM_BLUE)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + C4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					Cr_BLUE											0				Cb_BLUE											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

DCUx_LYR_CHRM_BLUE field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 Cr_BLUE	Cr coefficient for calculation of blue component.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 Cb_BLUE	Cb coefficient for calculation of blue component.

55.5.47 Compression Image Size Register (DCUx_COMP_IMSIZE)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + C8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										COMP_IMSIZE																					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

DCUx_COMP_IMSIZE field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–0 COMP_IMSIZE	Compressed Image size in bytes for RLE coded layer.

55.5.48 Update Mode Register (DCUx_UPDATE_MODE)

Address: Base address + CCh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MODE	READREG	0													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_UPDATE_MODE field descriptions

Field	Description
31 MODE	Do not set the MODE bit while the READREG is also set as this will block automatic updates. Do not set the MODE bit and the READREG register in the same write operation.

Table continues on the next page...

DCUx_UPDATE_MODE field descriptions (continued)

Field	Description
	The first frame transfer must be initiated manually using the READREG bit. Subsequent transfers can be performed automatically or manually. 0 Transfer of register values during vertical blanking period only when READREG is set 1 Automatic transfer of register values during vertical blanking period
30 READREG	When the MODE bit is clear this bit is a control bit which can be written to initiate a transfer of register value during the next vertical blanking period. 1'b1: (MODE=0) Transfer register values on next vertical blanking period. **NOTE:** If a transfer is underway, it is possible that changes to layer control descriptor values may be included in the transfer. 0 (MODE=0) No transfer scheduled. When the MODE bit is set, this bit is a status bit which indicates when a register transfer is underway. (MODE=1) No transfer is underway. 1 (MODE=0) Transfer register values on next vertical blanking period. (MODE=1) Register value transfer is underway.
29–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.49 Underrun Register (DCUx_UNDERRUN)

Address: Base address + D0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					LINE											0					PIXEL										
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_UNDERRUN field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 LINE	Line number where the underrun occured.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–0 PIXEL	Pixel number where the under run occured.

55.5.50 Global Protection Register (DCUx_GLBL_PROTECT)

Address: Base address + 100h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HLB	0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_GLBL_PROTECT field descriptions

Field	Description
31 HLB	Hard Lock Bit. This bit cannot be cleared once it is set by software. It can only be cleared by a system reset. 0 All SLB's are write protected and cannot be modified 1 All SLB's are accessible and can be modified
30–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.51 Soft Lock Bit Layer 0 Register (DCUx_SFT_LCK_BIT_L0)

Address: Base address + 104h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	SLB_L0_1	SLB_L0_2	SLB_L0_3	SLB_L0_4	0	0	0	0	SLB_L0_5	SLB_L0_6	SLB_L0_7	0
W	WEN_LO_1	WEN_LO_2	WEN_LO_3	WEN_LO_4					WEN_LO_5	WEN_LO_6	WEN_LO_7					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_SFT_LCK_BIT_L0 field descriptions

Field	Description
31 WEN_LO_1	Write Enable for Soft Lock Bit SLB_L0_1. 0 SLB is not modified 1 Value is written to SLB
30 WEN_LO_2	Write Enable for Soft Lock Bit SLB_L0_2. 0 SLB is not modified 1 Value is written to SLB
29 WEN_LO_3	Write Enable for Soft Lock Bit SLB_L0_3. 0 SLB is not modified 1 Value is written to SLB
28 WEN_LO_4	Write Enable for Soft Lock Bit SLB_L0_4. 0 SLB is not modified 1 Value is written to SLB

Table continues on the next page...

DCUx_SFT_LCK_BIT_L0 field descriptions (continued)

Field	Description
27 SLB_L0_1	Soft Lock Bit for Control Desc L0_1 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
26 SLB_L0_2	Soft Lock Bit for Control Desc L0_2 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
25 SLB_L0_3	Soft Lock Bit for Control Desc L0_3 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
24 SLB_L0_4	Soft Lock Bit for Control Desc L0_4 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
23 WEN_LO_5	Write Enable for Soft Lock Bit SLB_L0_5. 0 SLB is not modified 1 Value is written to SLB
22 WEN_LO_6	Write Enable for Soft Lock Bit SLB_L0_6. 0 SLB is not modified 1 Value is written to SLB
21 WEN_LO_7	Write Enable for Soft Lock Bit SLB_L0_7. 0 SLB is not modified 1 Value is written to SLB
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 SLB_L0_5	Soft Lock Bit for Control Desc L0_5 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
18 SLB_L0_6	Soft Lock Bit for Control Desc L0_6 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
17 SLB_L0_7	Soft Lock Bit for Control Desc L0_7 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
16–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.52 Soft Lock Bit Layer 1 Register (DCUx_SFT_LCK_BIT_L1)

Address: Base address + 108h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	SLB_L1_1	SLB_L1_2	SLB_L1_3	SLB_L1_4	0	0	0	0	SLB_L1_5	SLB_L1_6	SLB_L1_7	0
W	WEN_L1_1	WEN_L1_2	WEN_L1_3	WEN_L1_4					WEN_L1_5	WEN_L1_6	WEN_L1_7					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_SFT_LCK_BIT_L1 field descriptions

Field	Description
31 WEN_L1_1	Write Enable for Soft Lock Bit SLB_L1_1. 0 SLB is not modified 1 Value is written to SLB
30 WEN_L1_2	Write Enable for Soft Lock Bit SLB_L1_2. 0 SLB is not modified 1 Value is written to SLB
29 WEN_L1_3	Write Enable for Soft Lock Bit SLB_L1_3. 0 SLB is not modified 1 Value is written to SLB
28 WEN_L1_4	Write Enable for Soft Lock Bit SLB_L1_4. 0 SLB is not modified 1 Value is written to SLB

Table continues on the next page...

DCUx_SFT_LCK_BIT_L1 field descriptions (continued)

Field	Description
27 SLB_L1_1	Soft Lock Bit for Control Desc L1_1 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
26 SLB_L1_2	Soft Lock Bit for Control Desc L1_2 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
25 SLB_L1_3	Soft Lock Bit for Control Desc L1_3 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
24 SLB_L1_4	Soft Lock Bit for Control Desc L1_4 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
23 WEN_L1_5	Write Enable for Soft Lock Bit SLB_L1_5. 0 SLB is not modified 1 Value is written to SLB
22 WEN_L1_6	Write Enable for Soft Lock Bit SLB_L1_6. 0 SLB is not modified 1 Value is written to SLB
21 WEN_L1_7	Write Enable for Soft Lock Bit SLB_L1_7. 0 SLB is not modified 1 Value is written to SLB
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 SLB_L1_5	Soft Lock Bit for Control Desc L1_5 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
18 SLB_L1_6	Soft Lock Bit for Control Desc L1_6 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
17 SLB_L1_7	Soft Lock Bit for Control Desc L1_7 Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
16–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.53 Soft Lock Display Size Register (DCUx_SFT_LCK_DISP_SIZE)

Address: Base address + 10Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0			SLB_DISP	0										
W	WEN_DISP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_SFT_LCK_DISP_SIZE field descriptions

Field	Description
31 WEN_DISP	Write Enable for Soft Lock Bit SLB_DISP. 0 SLB is not modified 1 Value is written to SLB
30–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SLB_DISP	Soft Lock Bit for DISP_SIZE Register. This bit cannot be cleared once set by software. Can only be cleared by system reset. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
26–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.54 Soft Lock Hsync/Vsync Parameter Register (DCUx_SFT_LCK_HS_VS_PARA)

Address: Base address + 110h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0		SLB_HSYNC	SLB_VSYNC	0									
W	WEN_HSYNC	WEN_VSYNC														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_SFT_LCK_HS_VS_PARA field descriptions

Field	Description
31 WEN_HSYNC	Write Enable for Soft Lock Bit SLB_HSYNC. 0 SLB is not modified 1 Value is written to SLB
30 WEN_VSYNC	Write Enable for Soft Lock Bit SLB_VSYNC 0 SLB is not modified 1 Value is written to SLB
29–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SLB_HSYNC	Soft Lock Bit for HSYNC Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
26 SLB_VSYNC	Soft Lock Bit for VSYNC Register.

Table continues on the next page...

DCUx_SFT_LCK_HS_VS_PARA field descriptions (continued)

Field	Description
	0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
25–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.55 Soft Lock POL Register (DCUx_SFT_LCK_POL)

Address: Base address + 114h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0			SLB_POL	0										
W	WEN_POL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_SFT_LCK_POL field descriptions

Field	Description
31 WEN_POL	Write Enable for Soft Lock Bit SLB_POL 0 SLB is not modified 1 Value is written to SLB
30–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SLB_POL	Soft Lock Bit for SYN_POL Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
26–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.56 Soft Lock L0 Transparency Register (DCUx_SFT_LCK_L0_TRANSP)

Address: Base address + 118h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0		SLB_L0_FCOLOR	SLB_L0_BCOLOR	0									
W	WEN_L0_FCOLOR	WEN_L0_BCOLOR														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_SFT_LCK_L0_TRANSP field descriptions

Field	Description
31 WEN_L0_ FCOLOR	Write Enable for Soft Lock Bit SLB_L0_FCOLOR 0 SLB is not modified 1 Value is written to SLB
30 WEN_L0_ BCOLOR	Write Enable for Soft Lock Bit SLB_L0_BCOLOR 0 SLB is not modified 1 Value is written to SLB
29–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DCUx_SFT_LCK_L0_TRANSP field descriptions (continued)

Field	Description
27 SLB_L0_ FCOLOR	Soft Lock Bit for L0_FCOLOR Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
26 SLB_L0_ BCOLOR	Soft Lock Bit for L0_BCOLOR Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
25–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.57 Soft Lock L1 Transparency Register (DCUx_SFT_LCK_L1_TRANSP)

Address: Base address + 11Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0		SLB_L1_FCOLOR	SLB_L1_BCOLOR	0									
W	WEN_L1_FCOLOR	WEN_L1_BCOLOR														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DCUx_SFT_LCK_L1_TRANSP field descriptions

Field	Description
31 WEN_L1_ FCOLOR	Write Enable for Soft Lock Bit SLB_L1_FCOLOR 0 SLB is not modified 1 Value is written to SLB
30 WEN_L1_ BCOLOR	Write Enable for Soft Lock Bit SLB_L1_BCOLOR 0 SLB is not modified 1 Value is written to SLB
29–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 SLB_L1_ FCOLOR	Soft Lock Bit for L1_FCOLOR Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
26 SLB_L1_ BCOLOR	Soft Lock Bit for L1_BCOLOR Register. 0 Associated protected register is not locked and writeable 1 Associated protected register is locked for write access
25–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

55.5.58 Control Descriptor Ln_0 Register (DCUx_CTRLDESCL*n*_1)

This register sets the height and width of the layer associated with the register.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 200h offset + (64d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					HEIGHT											0					WIDTH										
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- This register resides in RAM, so reset values are undefined.

DCUx_CTRLDESCL*n*_1 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DCUx_CTRLDESCL*n*_1 field descriptions (continued)

Field	Description
26–16 HEIGHT	Height of the layer in pixels.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–0 WIDTH	Width of the layer (in pixels). The layer width must be in multiples of the number of pixels that can be stored in 32 bits, and therefore differs depending on color encoding. For example, if 2 bits per pixel format is used, then the layer width must be configured in multiples of 16. See Layer size and positioning .

55.5.59 Control Descriptor Ln_1 Register (DCUx_CTRLDESCL*n*_2)

This register sets the origin (top/left) of the layer associated with the register.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 204h offset + (64d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				POSY												0				POSX											
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- This register resides in RAM, so reset values are undefined.

DCUx_CTRLDESCL*n*_2 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–16 POSY	Two's complement signed value setting the vertic al position of top row of the layer, where 0 is the top row of the panel. Positive values are below and negative values are above the top row of the panel.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–0 POSX	Two's complement signed value setting the horizontal position of left hand column of the layer, where 0 is the left-hand column of the panel. Positive values are to the right and negative values are to the left the left-hand column of the panel.

55.5.60 Control Descriptor Ln_2 Register (DCUx_CTRLDESCL*n*_3)

This register sets the beginning address of layer data.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 208h offset + (64d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ADDR																															
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

- This register resides in RAM, so reset values are undefined.

DCUx_CTRLDESCL*n*_3 field descriptions

Field	Description
31–0 ADDR	Address of layer data in the memory. The address programmed should be 64-bit aligned.

55.5.61 Control Descriptor Ln_3 Register (DCUx_CTRLDESCL*n*_4)

This register controls various graphics options and whether the layer is enabled.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 20Ch offset + (64d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	EN	TILE_EN	DATA_SEL	SAFETY_EN	TRANS								BPP			
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RLE_EN	LUOFFS											0	BB	AB	
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

- This register resides in RAM, so reset values are undefined.

DCUx_CTRLDESCL*n*_4 field descriptions

Field	Description
31 EN	Enable the layer. 0 OFF 1 ON
30 TILE_EN	Enable the Tile Mode. This bit is available only for layer numbers 0 through 7 and 56 through 63. For the rest of the layers this bit is reserved and returns zero. 0 OFF 1 ON
29 DATA_SEL	Selects the Tile data either from MCU memory or CLUT. This bit is available only for layer numbers 0 through 7 and 56 through 63. For the rest of the layers this bit is reserved and returns zero. 0 Tile Mode data resides in the MCU memory 1 Tile mode data resides in the CLUT
28 SAFETY_EN	Safety Mode Enable Bit. Valid only for layer 0 and layer 1. For registers of all other layers, this should be set to 0. 0 Safety Mode is disabled 1 Safety Mode is enabled for this layer
27–20 TRANS	Transparency Level. Specifies the alpha value for the layer. This value may be used by the blending engine to blend pixels on this layer. Value can vary between 0-255 iwhere 0 is completely transparent adn 255 is completely opaque.
19–16 BPP	Layer encoding format (bit per pixel) 0000 1 bpp 0001 2 bpp 0010 4 bpp 0011 8 bpp 100 16 bpp (RGB565) 0101 24 bpp 0110 32 bpp (ARGB8888) 0111 Transparency mode 4 bpp 1000 Transparency mode 8bpp 1001 Luminance offset mode 4 bpp 1010 Luminance offset mode 8 bpp 1011 16 bpp (ARGB1555) 1100 16 bpp (ARGB4444) 1101 16 bpp (APAL8 mode) 1110 YCbCr422 (the blend engine allows only a single YCbCr layer in any blend operation) 1111 Reserved
15 RLE_EN	Enable RLE mode for layer.

Table continues on the next page...

DCUx_CTRLDESCL*n*_4 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled
14–4 LUOFFS	Look Up Table offset. Value gives the offset to the start address of the CLUT or tile (when used in internal tile mode) in the CLUT/TILE RAM.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 BB	Chroma Keying. 0 OFF 1 ON
1–0 AB	Alpha Blending. 00 No alpha Blending 01 Blend only the pixels selected by chroma keying in case BB=1'b1 10 Blend the whole frame 11 Same functionality as 2'b00

55.5.62 Control Descriptor Ln_4 Register (DCUx_CTRLDESCL*n*_5)

This register sets the maximum Chroma Keying values for RGB.

Refer to Layer size and positioning for a description of Chroma Keying.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 210h offset + (64d × i), where i=0d to 63d

Bit	31–24	23–16	15–8	7–0
R	0	CKMAX_R	CKMAX_G	CKMAX_B
W				
Reset	0* 0* 0* 0* 0* 0* 0* 0*	0* 0* 0* 0* 0* 0* 0* 0*	0* 0* 0* 0* 0* 0* 0* 0*	0* 0* 0* 0* 0* 0* 0* 0*

* Notes:
- This register resides in RAM, so reset values are undefined.

DCUx_CTRLDESCL*n*_5 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DCUx_CTRLDESCL*n*_5 field descriptions (continued)

Field	Description
23–16 CKMAX_R	Chroma Keying Max Red Component.
15–8 CKMAX_G	Chroma Keying Max Green Component
7–0 CKMAX_B	Chroma Keying Max Blue Component.

55.5.63 Control Descriptor Ln_5 Register (DCUx_CTRLDESCL*n*_6)

This register sets the minimum Chroma Keying values for RGB.

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 214h offset + (64d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								CKMIN_R							
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CKMIN_G								CKMIN_B							
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- This register resides in RAM, so reset values are undefined.

DCUx_CTRLDESCL*n*_6 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 CKMIN_R	Chroma Keying Minimum Red Component
15–8 CKMIN_G	Chroma Keying Minimum Green Component.
7–0 CKMIN_B	Chroma Keying Minimum Blue Component.

55.5.64 Control Descriptor Ln_6 Register (DCUx_CTRLDESCL*n*_7)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 218h offset + (64d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					TILE_VER_SIZE										
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									TILE_HOR_SIZE						
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- This register resides in RAM, so reset values are undefined.

DCUx_CTRLDESCL*n*_7 field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26–16 TILE_VER_SIZE	Height of the TILE (in pixels). This bitfield is available only for layer numbers 0 through 0 and 56 through 63. For the rest of the layers, this bitfield is reserved and returns zero.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 TILE_HOR_SIZE	Width of the TILE (in multiples of 16 pixels). This bitfield is available only for layer numbers 1 through 8 and 57 through 64. For the rest of the layers, this bitfield is reserved and returns zero.

55.5.65 Control Descriptor Ln_7 Register (DCUx_CTRLDESCL*n*_8)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 21Ch offset + (64d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								FGn_FCOLOR																							
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- This register resides in RAM, so reset values are undefined.

DCUx_CTRLDESCL*n*_8 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–0 FGn_FCOLOR	Foreground color to use when the layer is configured to use a transparency mode.

55.5.66 Control Descriptor Ln_8 Register (DCUx_CTRLDESCL*n*_9)

NOTE

A write to this register does not take effect until the value is transferred to the frame timing logic - see Transfer of DCU Configuration .

Address: Base address + 220h offset + (64d × i), where i=0d to 63d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1								FGn_BCOLOR																							
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- This register resides in RAM, so reset values are undefined.

DCUx_CTRLDESCL*n*_9 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
23–0 FGn_BCOLOR	Background color to use when the layer is configured to use a transparency mode

55.6 Functional Description

The DCU4 is a master on the NIC; it fetches graphic source information directly from memory and dynamically performs blending and bit-bliting operations before delivering data to a TFT LCD panel.

55.6.1 Graphic sources

As the DCU4 is a master on the NIC, it can access directly any memory or device connected to the NIC as a slave. This includes all on-chip RAM, and any slave capable of providing high enough data rates, such as, for example an expanded bus interface or a QuadSPI module. Therefore, any compatible graphic stored anywhere on-chip or in an accessible interface can be displayed on the connected TFT LCD panel with no further intervention from the CPU, except to program the DCU4 to fetch and place it. The DCU4 also includes a dedicated memory to store the graphic for its cursor layer.

55.6.2 TFT LCD panel configuration

The nature and timing of the signals required by TFT LCD panels vary greatly between manufacturers. Therefore, the DCU4 allows highly flexible and detailed configuration of these signals.

Timing diagrams for TFT LCD panels are typically divided into a horizontal timing chart and a vertical timing chart. See Figure 55-1929 for details.

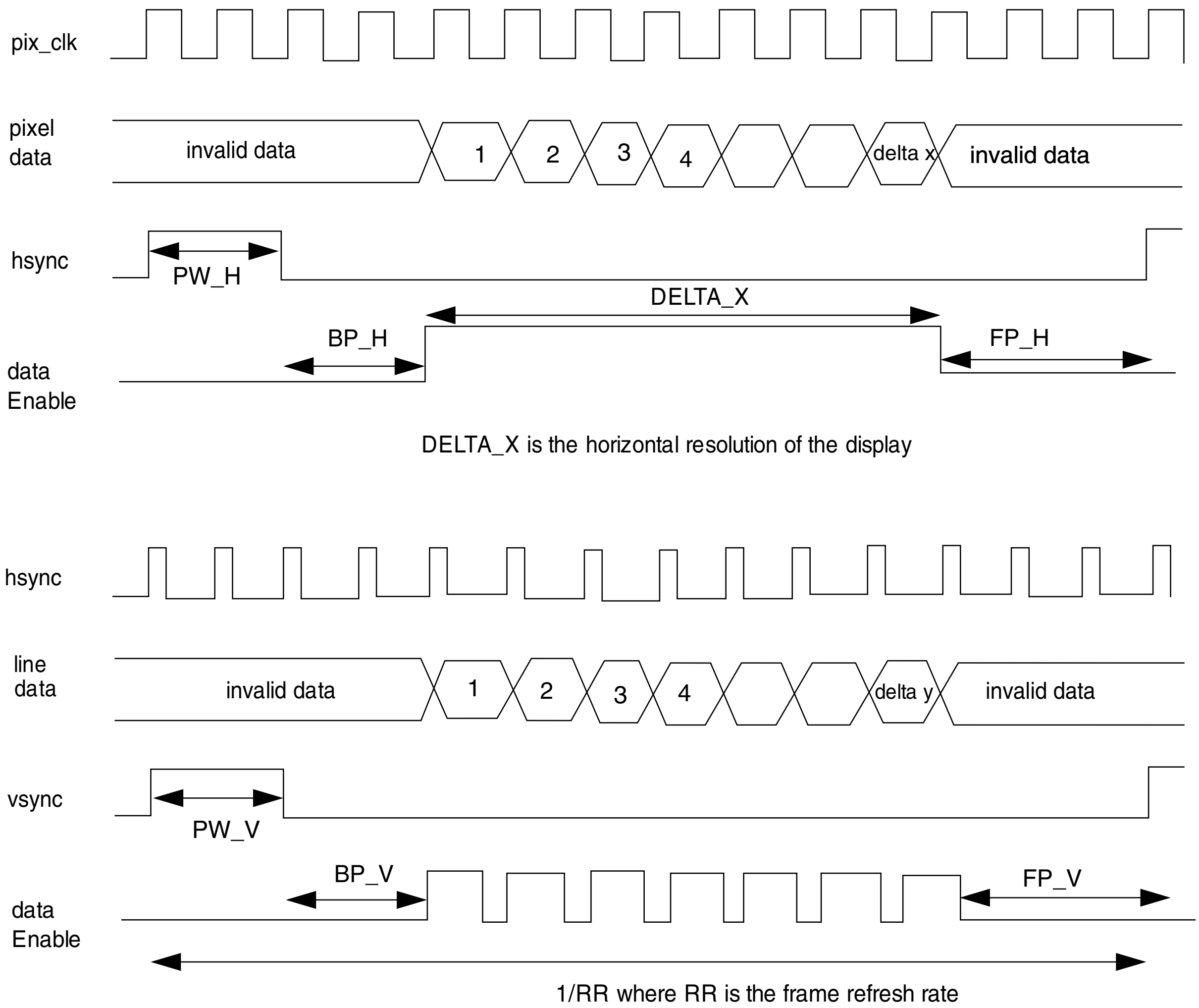


Figure 55-1929. HSYNC and VSYNC timing diagram

The number of pixel data slots in the horizontal timing diagram is defined by the width of the panel. The number of line data slots is defined by the height of the panel. Both of these values are defined in the DISP_SIZE register (DELTA_X, DELTA_Y). The width of the panel must always be defined as a multiple of 16.

The timing of the pixel clock is defined by the DIV_RATIO register and the frequency of the clock supplied to the DCU4.

In addition to defining the number and timing of pixels in each line and the number of lines, it is normal for TFT LCD panel manufacturers to define other timing signals in terms of pixel clock periods or of the number of horizontal lines. The DCU4 also follows this convention.

If the TFT LCD panel requires a horizontal synchronizing signal (HSYNC) and/or a data enable signal, then these can be configured using the fields in the HSYN_PARA register. HSYNC provides a pulse to give the panel notice that the next line of pixel data is about to start, and the data enable signal indicates when that data is present. The PW_H bit field

indicates the width of the HSYNC pulse, in pixel data clock periods. The BP_H bit field defines the delay between the end of the HSYNC pulse and the start of the data enable signal (and pixel data delivery), in pixel clock periods. The FP_H bit field defines the delay between the end of the data enable signal (and pixel data delivery) and the next HSYNC pulse, in pixel clock periods. FP_H and BP_H have minimum values of 1.

If the TFT LCD panel requires a vertical synchronizing signal (VSYNC), then this can be configured using the fields in the VSYN_PARA register. VSYNC provides a pulse to give the panel notice that the next frame of pixel data lines is about to start, and the panel defines delays before and after this pulse, in terms of pixel clock periods. The PW_V bit field indicates the width of the VSYNC pulse in horizontal line periods. The BP_V bit field defines the delay between the end of the VSYNC pulse and the start of the next pixel data (data enable signal), in horizontal line periods. The FP_V bit field defines the delay between the end of the last pixel data (data enable signal) and the next VSYNC pulse, in horizontal line periods. FP_V and BP_V have minimum values of 1.

The polarity of all these signals, including the pixel data itself, may be inverted by using the control bits in the SYN_POL register.

The refresh rate for the panel can be calculated using Table 55-1931 and Pixel Clock calculation below.

$$RR = \frac{\text{pix\_clk}}{(\text{DELTA\_}X+\text{FP\_}H+\text{PW\_}H+\text{BP\_}H)\times(\text{DELTA\_}Y+\text{FP\_}V+\text{PW\_}V+\text{BP\_}V)}$$

where:	pix_clk is the pixel clock
	DELTA_X is the horizontal resolution (in pixels)
	DELTA_Y is the vertical resolution (in pixels)
	FP_H is the hsync front porch pulse width (in pixel clock cycles)
	BP_H is the hsync back porch pulse width (in pixel clock cycles)
	PW_H is the hsync active pulse width (in pixel clock cycles)
	FP_V is the vsync front porch pulse width (in number of horizontal lines)
	BP_V is the vsync back porch pulse width (in number of horizontal lines)
	PW_V is the vsync active pulse width (in number of horizontal lines)

pix_clk = DCU4 Clock / (DIV_RATIO – 1),

where DIV_RATIO is an integer value in the DIV_RATIO register that can range from 1 to 128.

The configuration values in these registers are "locked-in" for the panel when a frame refresh cycle is initiated. For more information about this process see Transfer of DCU Configuration.

55.6.3 DCU4 Mode selection and background color

Once the DCU4 is configured for use with a particular TFT LCD panel, it can be enabled for use. There are five modes to choose from, as shown in the following table.

Table 55-1932. List of DCU operating modes

Mode	DCU_MODE[1:0]	PDI_EN	Description
Off	00	X	DCU4 disabled; the TFT LCD panel is not driven.
Color bar	11	X	DCU4 displays a test pattern consisting of vertical bands of programmable color.
Normal	01	0	DCU4 blends layers and displays result on TFT LCD panel.
PDI normal	01	1	As normal mode, except that the panel timing is defined by the input on the PDI interface, and the background color is replaced by the content provided on the PDI interface.
PDI slave	01	0	The DCU4 synchronizes its timing to an external signal when PDI_SLAVE_MODE is enabled.

The DCU_MODE, PDI_EN and PDI_SLAVE_MODE control bits are in the DCU_MODE register. The DCU4 has an interface enable bit for the TFT LCD panel interface called RASTER_EN, also in the DCU_MODE register. When RASTER_EN is 0, the raster scanning of pixels to the panel is disabled, but the pixel clock continues to run if enabled on the I/O pin.

Color bar mode is intended for testing the interface between the DCU4 and the TFT LCD panel. In this mode, the panel is divided into eight vertical strips of equal width, and the strips display a single color whose RGB value is specified in the COLBAR_1 to COLBAR_8 registers. At reset, the colors are set to black, blue, cyan, green, yellow, red, magenta, and white, where positive logic for the RGB values is assumed. The mode can be used to verify correct connection of the interface to the DCU4 and correct timing configuration of the interface. In this mode, any layer configuration settings are ignored.

In Normal mode, the DCU4 operates according to the timings specified in TFT LCD panel configuration" and displays graphics according to the configuration of its layers. The BGND register sets the RGB color of the background shown when no other layers are present. This background color is included in the layer blending process but, since it is always the background, it does not include any layer blending settings.

In PDI normal mode, the DCU4 adopts the timing provided on the PDI interface and replaces the background color by the pixel data coming from the PDI interface. The timing values set in the DCU4 are ignored in this mode, and the pixel clock and

synchronization signals are taken from the PDI interface and passed to the TFT LCD panel. The content of the panel is a combination of the incoming pixel stream and layers generated by the DCU4.

PDI slave mode allows the DCU4 to synchronize with the external timing signals on the PDI input.

55.6.4 Layer configuration and blending

Users control the graphical content of the TFT panel by manipulating the configuration of elements in the DCU4 called layers. Each layer has control descriptors that define the size, position, memory encoding, blending, and memory location of the graphic to be displayed. The DCU4 provides 64 independent layers that all display graphics with a fixed priority, and this affects how individual pixels are blended when layers overlap. The blending setting on each layer allows the pixels on that layer to be opaque, partially transparent, or fully transparent, which allows them to combine with pixels on other layers that they overlap.

55.6.4.1 Blending priority of layers

The 64 layers available in the DCU4 are each fixed in priority order, with layer 0 being the highest priority, layer 1 being the second highest priority, and so on until layer 63, which is the lowest priority. The priority is used by the DCU4 to define how to blend individual pixels within the layers. For example, if layer 0 is defined as not being blended with other layers and a pixel on layer 0 overlaps a pixel on layer 1 then the pixel on layer 0 will be visible on the panel unchanged by the pixel on layer 1. However, if layer 0 is defined as being partially transparent, then the DCU4 will blend the overlapping pixel such that the result is a combination of the pixel on layer 0 and the pixel on layer 1. It is possible to blend up to six layers at each pixel position.

As there is a maximum number of layers that can blended together, then any pixel on a layer that is lower than the threshold priority will not be included in any blend. If a pixel is on a layer that has the lowest priority in any blending scheme, then the blending settings for that pixel are ignored and the pixel is treated as a background pixel. This means that a lower priority layer may have some pixels completely obscured by those on higher priority layers on one part of the panel, and some other pixels visible or blended on other parts of the panel.

Figure 55-1930 shows how the pixel blend takes place inside the DCU4. The priority of the layers determines at which stage of the blend the pixel enters. Any pixels lower than the threshold priority are ignored and, as can be seen, the blend settings for the lowest

priority pixel is also ignored. The maximum number of pixels in the blend is configured by the BLEND_ITER bit field in the DCU_MODE register. As can be seen in the figure, the blending process is iterative so that six-pixel blending takes more DCU4 clock cycles than five-pixel blending, and three-pixel blending takes more DCU4 clock cycles than two-pixel blending.

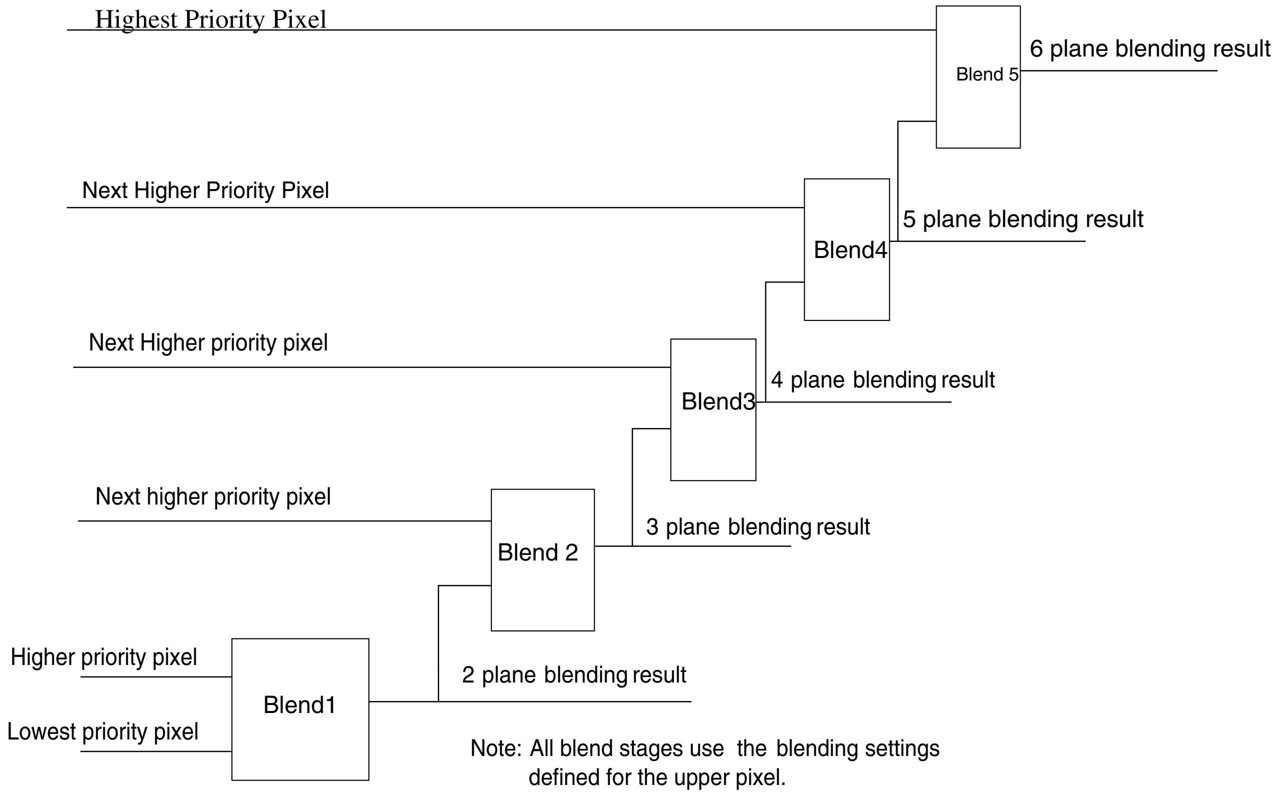


Figure 55-1930. Pixel blending stack

The number of pixels in each blend depends on two factors: The BLEND_ITER setting which defines the maximum and the actual number of layers present at the particular location on the panel. If fewer layers are present than the value in BLEND_ITER then only the number of blend stages required will occur. If more layers are present than the value in BLEND_ITER then the pixels below the lowest priority pixel are ignored. Note that the blend stack always operates from the lowest to highest such that a two layer blend will always use FIFO CH1 and CH2 to produce the 2 pixel blending result whereas a five layer blend will use CH1, CH2, CH3, CH4 and CH5 to produce the 5 pixel blending result.

This priority concept is illustrated in Figure 55-1931 and Figure 55-1932. In this case, there are five layers enabled, and each contains a graphic that is a solid rectangular block of a single color. The size and shape of each layer is different. The background color of the panel is set to grey and layers have been placed such that they overlap each other.

Figure 55-1931 shows the individual source graphics and the case where no layer has any blending enabled. Here, the highest priority layer (in this case layer 0) is fully visible. Layer 1 is visible where layer 0 does not overlap it. Layer 2 is visible where layer 1 does not overlap it. Layer 3 is overlapped by layers 0 and 1 and so is only partially visible. Layer 4 is partially obscured by all of the other layers. Note that layer 4 is higher priority than the background color.

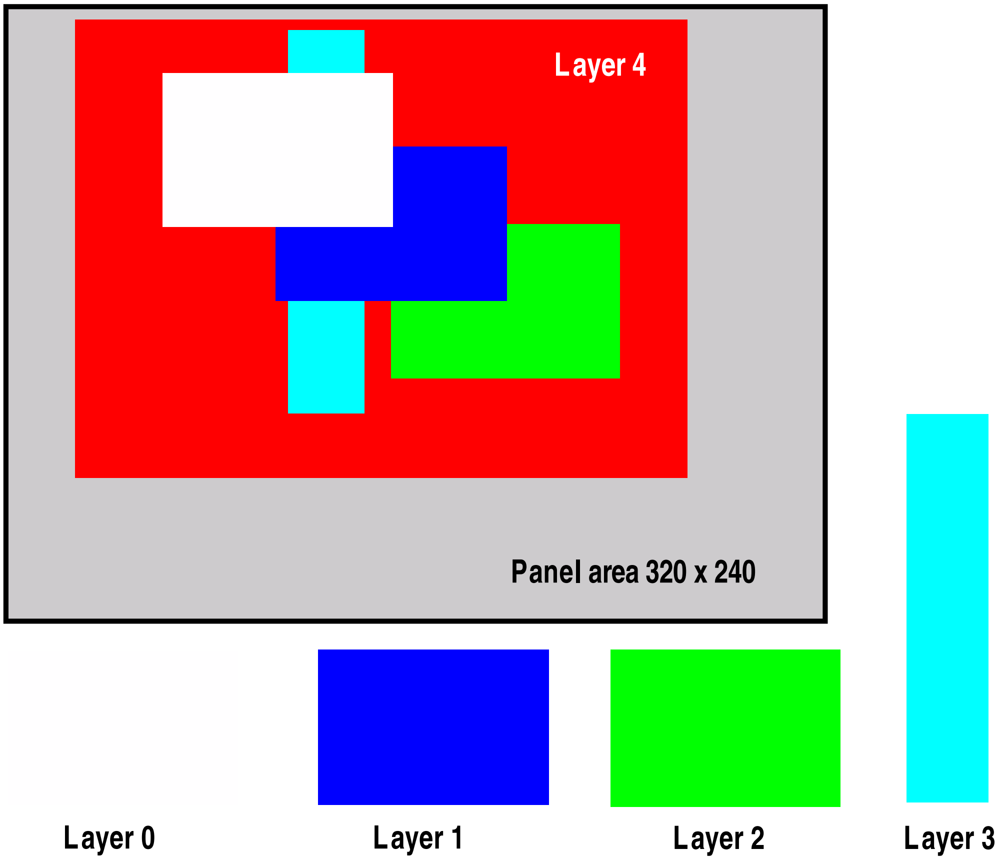


Figure 55-1931. Example of layer placement with no blending

Figure 55-1932 shows the same layer configuration except that, in this case, the layers have been made 50% transparent and the depth of the pixel blend is set to 3. The pixels in layer 0 are now blended with pixels in the underlying layers. In particular, note region A where layer 0 is blended with layer 4 and the background color. This blending effect is repeated across all of the layers; however, note the pixels in region B. In this region the pixels from layer 0 are blended with those on layer 1 and layer 2; however, the pixels from layer 4 and the background color are not included in this blend. This is because the DCU4 is configured to blend three layers only, and so the blend setting for layer 2 is ignored for those pixels in region B.

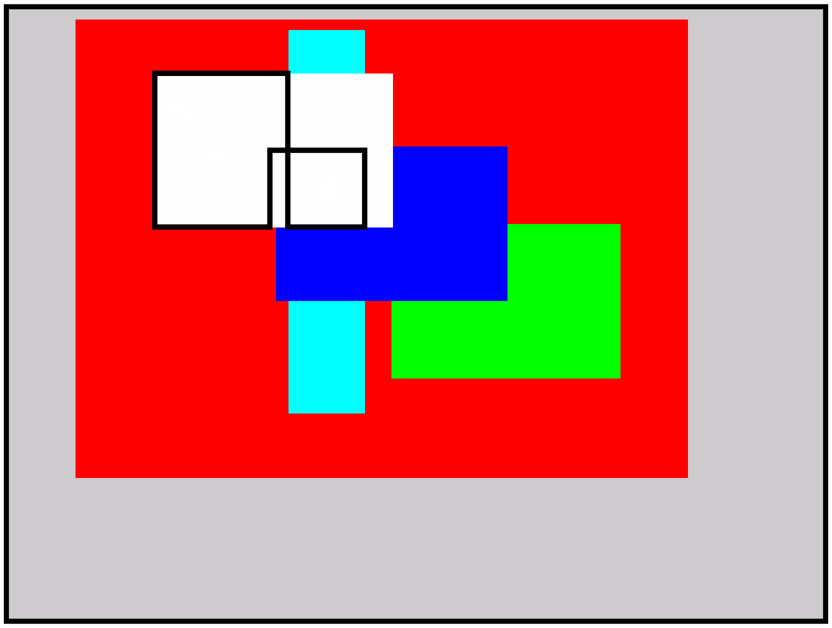


Figure 55-1932. Example of layer placement with 3-layer blending

All blending is performed using full 8-bits-per-component colors. The DCU4 automatically performs a color promotion on source data that is stored in less than RGB888 color.

55.6.4.2 Transfer of DCU Configuration

The configuration of the DCU occurs in two stages. Firstly the configuration values are written into the appropriate register by the user software. Secondly, these values are transferred into the frame timing logic for use on the next frame refresh. This means that the values written into the registers do not take effect until the transfer completes. Similarly, error conditions will not be detected until the transfer occurs and so error flags will not be set until the transfer is complete. Refer to the register descriptions for details of which registers require this transfer operation.

The transfer itself always occurs at a fixed time in relation to the vertical blanking period. The time at which the control descriptors are updated and ready to transfer is under user control and there are two control bits in the UPDATE_MODE register that determine how the DCU behaves. The READREG bit causes a single transfer to begin at the next frame blanking period. This bit is cleared when the transfer is complete. The MODE bit forces the transfer to occur on every frame blanking period. This means that any changes to the affected registers must be complete before the transfer is ready to begin otherwise coherency issues can occur. The DCU provides a status flag called PROG_END bit in the

INT_STATUS register which is asserted when the transfer is beginning and a status flag called LYR_TRANS_FINISH which indicates when the transfer is complete. Both of these status flags can generate an interrupt.

Changes to register contents before the transfer is complete may be copied as part of the transfer.

NOTE

Every transfer must be performed using the READREG bit until the DCU_MODE[DCU_MODE] bit field is set to non-zero. All subsequent transfers may be performed using either the READREG or the MODE bit method.

55.6.4.3 Control Descriptors

The control descriptor for each layer consists of nine registers, and all 64 control descriptors have common features that allow the display and blending of a graphic in the frame. In addition, the top two layers (0 and1) have additional control bits for the safety mode. The top and bottom eight layers (0 to 7 and 56 to 63) also have support for tile mode.

The control descriptors contain the configuration that will be used for a future frame. The current frame configuration will remain the same until all of the control descriptor contents are updated by the user and subsequently transferred to the frame timing logic. Once this transfer is complete the control descriptors can be modified for the next frame.

55.6.4.4 Layer size and positioning

The size of each layer is defined by register 1 in the control descriptor for the layer (CTRLDESCLn_1, where n is the layer number). The register contains two bit fields, HEIGHT and WIDTH, which determine the size and shape of the layer. Both fields are expressed in terms of the number of pixels in each dimension.

The HEIGHT bit field may take any value.

The WIDTH field has a restriction on the value it can take, depending on the data format of the graphic specified by the layer. This field must always be an integer multiple of the number of pixels that are represented by a 32-bit word except in the special case of 1 bit per pixel where the multiple is 16. The data format can range from 1 bit per pixel to 32 bits per pixel and so there is a range of multiples from 1 to 32. Table 55-1933 shows the multiples for the WIDTH bit field and some correct values.

Table 55-1933. Example of WIDTH multiples for different graphic data formats

Data format	WIDTH multiples	Example values
1 bpp	16	16, 32, 48, 64, …
2 bpp	16	16, 32, 48, 64, …
4 bpp	8	8, 16, 24, 32, …
8 bpp	4	4, 8, 12, 16, …
16 bpp	2	2, 4, 6, 8, …
24 bpp	4 (= 3 whole 32-bit words)	4, 8, 12, 16
32 bpp	1	1, 2, 3, 4, …
YUV422	4	4, 8, 12, 16, …

If the WIDTH bit field is set to an invalid multiple, then the layer configuration is invalid, the layer cannot be made visible, and an error flag is set in the layer parameter error register (PARR_ERR_STATUS1 and PARR_ERR_STATUS2).

The position of each layer on the panel is defined by register 2 in the control descriptor for the layer (CTRLDESCLn_2, where n is the layer number). The register contains two bit fields, POSY and POSX, which determine the location of the upper left pixel of the layer in the x and y axes. Both fields are expressed in terms of the number of pixels in each axis.

There are no restrictions on layer placement. Any layer can be placed and moved to any panel position. If a layer is placed so that pixels would appear beyond the dimensions of the panel, then the DCU4 displays the pixels on the panel and ignores the pixels off the panel.

55.6.4.5 Graphics and data format

The memory location of the graphic that is displayed on the layer is defined by register 3 in the control descriptor for the layer (CTRLDESCLn_3, where n is the layer number). This 32-bit value can contain the address of any memory location in the memory map of the MCU.

The format of the data that describes the graphic is defined by the BPP bit field in register 4 in the control descriptor for the layer (CTRLDESCLn_4, where n is the layer number). This value also influences the range of values for the width of the layer (see Layer size and positioning"). By choosing an appropriate format, it is possible to optimize the memory required by the graphics in use.

There are five modes where the RGB values of the pixels are stored directly in the graphic. In these modes, the DCU4 treats the data as describing a true RGB color. The modes are:

- ARGB8888, where the data defines 8-bit values for the red, green, blue, and alpha components of the image. This blends as ARGB, however, in this format the order of the bytes is reversed compared to other formats.
- RGB888, where the data defines 8-bit values for the red, green, and blue components of the image.
- RGB565 where the data defines 5-bit values for the red and blue components, and 6-bit values for the green component of the image.
- ARGB1555 where the data defines 5-bit values for the red, green, and blue components, and a 1-bit value for the alpha channel of the image.
- ARGB4444, where the data defines 4-bit values for the red, green, blue, and alpha components of the image.

The three 16-bit formats (RGB565, ARGB1555, and ARGB4444) are promoted to full 8 bit per component format by shifting the bits left so that the MSB of the component in the 16-bit format becomes the MSB of the 24/32 bpp (bit per pixel) format, and the LSB is filled with the value of the MSBs. For example, an RGB565 value of 10000:010000:11011 becomes 10000100:01000001:11011110. An RGB4444 value of 1010:0011:1100:0101 becomes 10101010:00110011:11001100:01010101. An RGB1555 value of 1:10100:01000:11011 becomes 11111111:10100101:01000010:11011110.

There are five indexed color formats (1/2/4/8 bpp & APAL8) where the data in the graphic does not define the RGB color to display. Instead, the data defines the entry in a color look-up table (CLUT) that contains a palette of ARGB colors. The maximum number of colors in the CLUT is defined by the size of the data stored in the graphic. For 1 bpp graphics, there is a maximum of two colors in the CLUT. For 2 bpp, there is a maximum of four colors. For 4 bpp and 8 bpp data, the maximums are 16 and 256 colors, respectively. In APAL8 mode(16 bpp), the upper 8 bits define the alpha component of the pixel and the lower 8 bits define the offset in the CLUT (the alpha component in the CLUT color is ignored).

The address of the first value in the CLUT is defined in the LUOFFS bit field of register 4 and the CLUT is the RAM block dedicated to the DCU4 which is described in CLUT/Tile RAM". Since the RGB values stored in the CLUT are 32-bit RGB, there is no need for further adjustment before blending.

The DCU4 also supports graphics encoded using luminance and chrominance format. This format is generically known as YUV and stores the luminance (brightness, Y) of a pixel separate from its chrominance (color information, U and V). This format is widely used by cameras and is supported by the PDI for direct video in as well as the DCU4 when stored in memory for display on a layer. The specific implementation used by the DCU4 is more accurately described as YCbCr422 which uses twice as many bits to describe the luminance as to describe the blue (Cb) and red (Cr) difference of the chrominance.

The DCU4 takes these pixels and converts them to RGB format using equations configured using its LYR_LUMA_COMP, LYR_CHROMA_RED, LYR_CHROMA_GREEN, and LYR_CHROMA_BLUE registers. The YCbCr format specifies a common chroma setting for two pixels; however, it is possible to interpolate the chroma for the pixels rather than setting both to the same value. This feature is enabled by the LYR_INTPOL_EN[EN] field. Due to the additional conversion step required, the DCU4 is able to blend a maximum of one layer encoded in YCbCr for each pixel. See “PDI YCbCr mode and DCU4 YCbCr color format.

There are four additional modes defined by the BPP bit field. These configure the graphic in transparency mode and luminance mode (see Transparency mode and blending" and Luminance mode" respectively).

There is a set storage format for each data format provided by the DCU4. These formats can be seen in the following tables.

Table 55-1934. Data Layout for ARGB8888

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	A0	R0	G0	B0	A1	R1	G1	B1
0x08	A2	R2	G2	B2	A3	R3	G3	B3

In the following table, the YCbCr422 format encodes chroma information across two pixels. Therefore, the chroma values apply to the even pixel denoted in the table and its adjacent odd pixel.

Table 55-1935. Data layout for YCbCr422 format

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	Y1	Cr0	Y0	Cb0	Y3	Cr2	Y2	Cb2
0x08	Y5	Cr4	Y4	Cb4	Y7	Cr6	Y6	Cb6

Table 55-1936. Data Layout for 24 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	B1	R0	G0	B0	G2	B2	R1	G1
0x08	R3	G3	B3	R2	B5	R4	G4	B4

For 16 bpp, data expected is in the form of RGB565, ARGB1555 or ARGB4444, or APAL8.

Table 55-1937. Generic data layout for 16 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p1[15:8]	p1[7:0]	p0[15:8]	p0[7:0]	p3[15:8]	p3[7:0]	p2[15:8]	p2[7:0]
0x08	p5[15:8]	p5[7:0]	p4[15:8]	p4[7:0]	p7[15:8]	p7[7:0]	p6[15:8]	p6[7:0]

Table 55-1938. Data layout for RGB565 format

Address offset	[31:27]	[26:21]	[20:16]	[15:11]	[10:15]	[4:0]
0x00	R1	G1	B1	R0	G0	B0
0x04	R3	G3	B3	R2	G2	B2

Table 55-1939. Data layout for ARGB4444 format

Address offset	[31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
0x00	A1	R1	G1	B1	A0	R0	G0	B0
0x04	A3	R3	G3	B3	A2	R2	G2	B2

Table 55-1940. Data layout for ARGB1555 format

Address offset	[31]	[30:26]	[25:21]	[20:16]	[15]	[14:10]	[9:5]	[4:0]
0x00	A1	R1	G1	B1	A0	R0	G0	B0
0x04	A3	R3	G3	B3	A2	R2	G2	B2

Table 55-1941. Data layout for APAL8 format

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	A1	offset 1	A0	offset 0	A3	offset 3	A2	offset 2
0x08	A5	offset 5	A4	offset 4	A7	offset 7	A6	offset 6

Table 55-1942. Data Layout for 8 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p3	p2	p1	p0	p7	p6	p5	p4
0x08	p11	p10	p9	p8	p15	p14	p13	p12

Table 55-1943. Data layout for 4 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p7-p6	p5-p4	p3-p2	p1-p0	p15-p14	p13-p12	p11-p10	p9-p8
0x08	p23-p22	p21-p20	p19-p18	p17-p16	p31-p30	p29-p28	p27-p26	p25-p24

Table 55-1944. Data Layout for 2 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p15-p12	p11-p8	p7-p4	p3-p0	p31-28	p27-24	p23-20	p19-16
0x08	p47-p44	p43-p40	p39-p36	p35-p32	p63-p60	p59-p56	p55-p52	p51-p48

Table 55-1945. Data Layout for 1 bpp

Address offset	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]
0x00	p63-p56	p55-48	p47-40	p39-p32	p31-p24	p23-p16	p15-p8	p7-p0
0x08	p127-p120	p119-p112	p111-p104	p103-p96	p95-p88	p87-p80	p79-p72	p71-p64

The DCU4 includes a flag that indicates when it has completed fetching graphics from memory for the current frame refresh. If required, this flag (DMA_TRANS_FINISH in the INT_STATUS register) can be used to determine when changes can be made to the source graphic content.

55.6.4.6 Alpha and Chroma-key blending

The blending configuration of each layer is defined by the BB, AB, and TRANS bit fields in register 4 in the control descriptor for the layer (CTRLDESCLn_4, where n is the layer number). The pixels affected by the blending configuration can be further selected by registers 5 and 6 in the control descriptor (CTRLDESCLn_4 and CTRLDESCLn_5). Depending on the priority and placement of the layer (see Blending priority of layers"), these bit fields and registers define how pixels from different layers are blended together.

The AB and BB bit fields define whether blending is active and whether the whole graphic or a selected portion is blended. Registers 5 and 6 specify the range of RGB colors that define the selected pixels. The TRANS bit field defines the transparency of the selected pixels.

The BB bit field defines whether the whole graphic, or only certain pixels, should be blended. When this bit is set, pixels that have an RGB value that falls into the range defined by registers 5 and 6 are considered to be selected and treated differently to the non-selected pixels in the graphic. This is a process known as chroma-keying since it is the color of the pixel that defines the selection. The selected pixels must be within the range defined by each color component of registers 5 and 6. See Table 55-1946 for examples of pixels that are selected and not selected when the given range is defined as 0x0080C0 to 0x0FB0FF.

Table 55-1946. Example of how chroma-key range selects pixels

Source pixel	Red 00–0F	Green 80–B0	Blue C0–FF	Comment
0x000000	P	X	X	Not selected
0x08C0C0	P	X	P	Not selected
0x08A0C0	P	P	P	Pixel is selected

The AB bit field defines how any selected and non-selected pixels are blended. By combining this control with the BB bit field it is possible to define 11 unique ways of blending the pixels on a layer dependent on the type of layer. Depending on the configuration defined by the AB and BB bit fields, the TRANS bit field combines the two pixels in every blend stage using the alpha value of the upper pixel (which has the effect of making this pixel more or less transparent and revealing more or less of the lower pixel).

The result of each blend stage is calculated for all three color components as shown in Figure 55-1932.

$$A = (\mathrm{BGPixel} * (255 - \mathrm{alpha})) + (\mathrm{FGPixel} * \mathrm{alpha})$$

The result of the calculation must then be divided by 255 to normalize the result. This calculation is performed as follows:

1. First Division output_val = A + (A >> 8)
2. Rounding off first addition & division if (((A>>7) & 0x1) == 0x1) output_val ++
3. Second Division with rounding output_val = output_val >>7; if ((output_val & 0x1) == 0x1)) output_val = output_val + 0x2; output_val = output_val >> 1

The blend can apply to pixels with no alpha channel (RGB) or with an alpha channel (ARGB) in different ways.

Table 55-1947 defines how the settings of the BB and AB bit fields affect the pixels in the layer; RGB modes are 1 bpp, 2 bpp, 4 bpp, 8 bpp, RGB565, and RGB888; ARGB modes are ARGB1555, ARGB4444, and BGRA8888.

Table 55-1947. Blend options for BB and AB configurations

Case	BB	AB[1:0]	Mode	Function
1	0	00	RGB	No blending, underlying pixels are obscured
2	1	00	RGB	Selected pixels are completely removed
3	0	01	RGB	The value in TRANS becomes the alpha channel of all pixels on the layer
4	1	01	RGB	The value in TRANS becomes the alpha channel of the selected pixels on the layer
5	0	10	RGB	Same as case 3
6	1	10	RGB	Selected pixels are completely removed and the value in TRANS becomes the alpha channel of the non-selected pixels on the layer
7	0	11	RGB	Reserved
8	1	11	RGB	Reserved
9	0	00	ARGB	No blending, pixel alpha is ignored and underlying pixels are obscured
10	1	00	ARGB	Selected pixels are completely removed, pixel alpha is ignored
11	0	01	ARGB	Pixel alpha is used to blend layer with underlying pixels. Value in TRANS is ignored.
12	1	01	ARGB	Uses the pixel alpha of the selected pixels only to blend layer with underlying pixels. Value in TRANS is ignored.
13	0	10	ARGB	The value in TRANS is multiplied with the pixel alpha value and the resultant alpha is used to blend all the pixels
14	1	10	ARGB	Selected pixels are completely removed, the value in TRANS is multiplied with the pixel alpha value and the resultant alpha is used to blend the non-selected pixels on the layer
15	0	11	ARGB	Reserved
16	1	11	ARGB	Reserved

Figure 55-1933 to Figure 55-1941 illustrate the effect of the cases identified in Table 55-1947. In all cases there is a single active layer and a white background color.

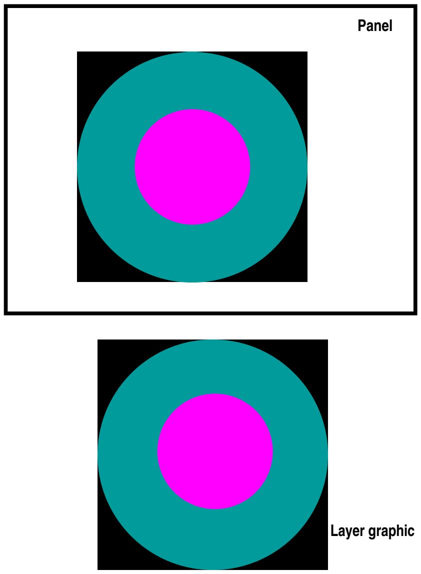


Figure 55-1933. Case 1 example (no blend)

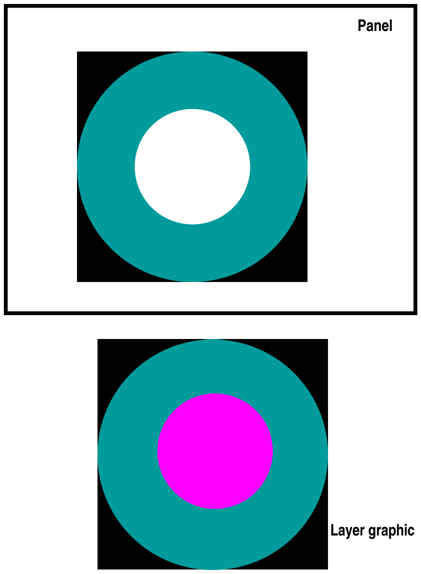


Figure 55-1934. Case 2 example (remove selected pixels)

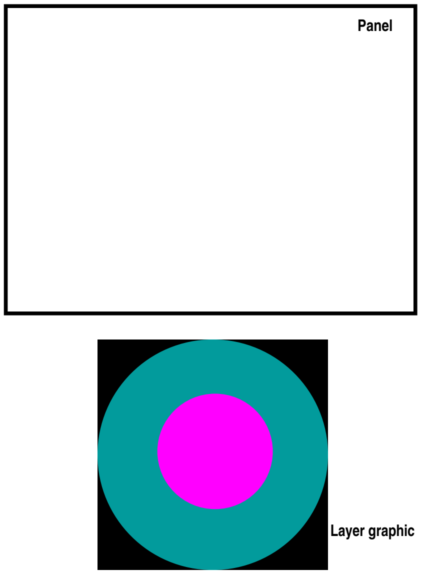


Figure 55-1935. Case 3 example (all pixels transparent)

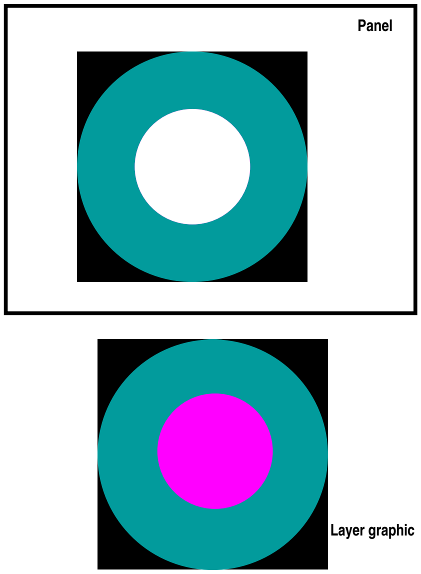


Figure 55-1936. Case 4 example (selected pixels transparent)

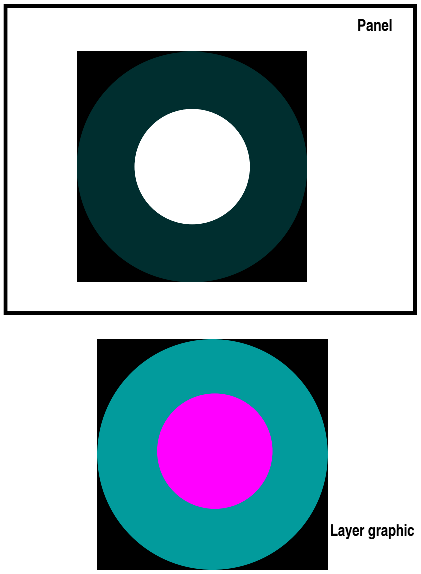


Figure 55-1937. Case 6 example (selected pixels removed, others transparent)

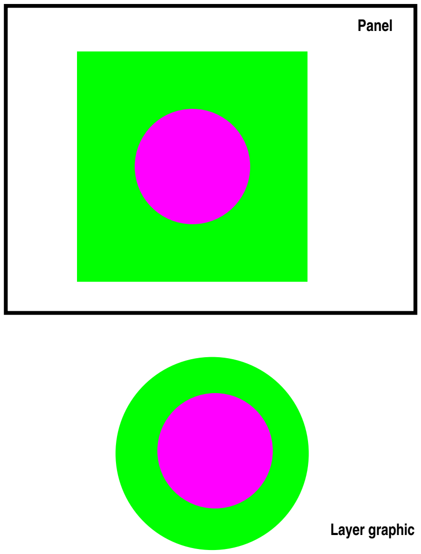


Figure 55-1938. Case 9 example (no blend, pixel alpha ignored)

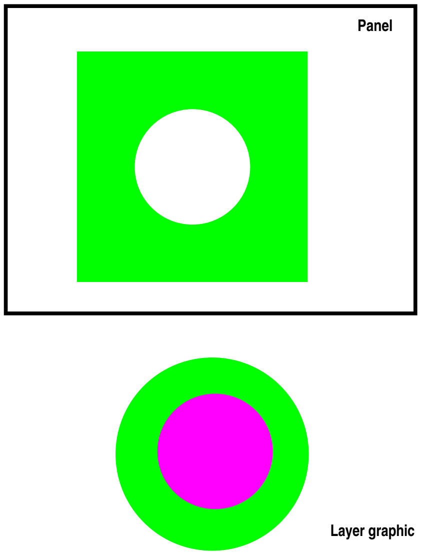


Figure 55-1939. Case 10 example (selected pixels removed, pixel alpha ignored)

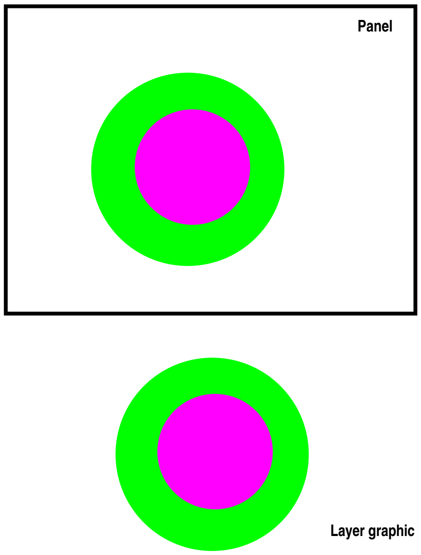


Figure 55-1940. Case 13 example (pixel and layer alpha used in blend)

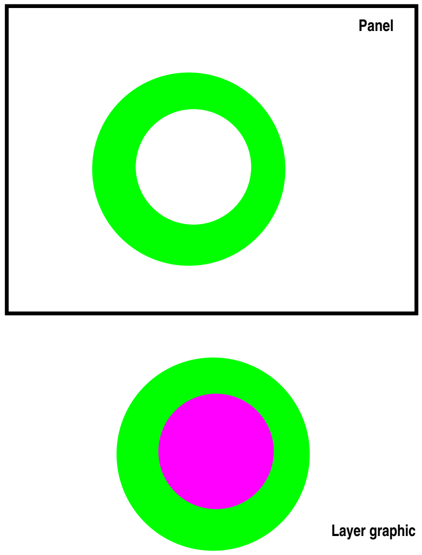


Figure 55-1941. Case 14 example (selected pixels removed, pixel and layer alpha used in blend)

55.6.4.7 Transparency mode and blending

Transparency mode is a special case for the graphic data format and is defined by the BPP bit field in register 4 in the control descriptor for the layer (CTRLDESCLn_4, where n is the layer number). This value also influences the range of values for the width of the layer (see Layer size and positioning"). By choosing an appropriate format, it is possible to optimize the memory required by the graphics in use.

In transparency mode, the source graphic does not contain any direct or indexed color information. Instead, the graphic data represents the alpha channel of the graphic. The DCU4 creates the final graphic by pre-blending a foreground color and background color using the alpha value of each pixel. The result of this pre-blend can then be blended with pixels on other layers using the normal blending process. Each layer has dedicated registers to contain the foreground and background colors for this mode within the control descriptor. These are registers CTRLDESCLn_8 and CTRLDESCLn_9, where n is the layer number. See Figure 11-76.

Transparency mode is typically used when a graphic must blend smoothly into the underlying layers, but where a rich color palette is not required. Examples include text where this mode allows the text to blend smoothly with any background — this is known as anti-aliasing.

There are two transparency modes available: 4 bpp and 8 bpp. The 4 bit data is translated to 8 bit by concatenating the 4 bits.The result of the pre-blend can be treated as an RGB888 graphic and blended in a similar way to previously described, or it can be treated as a special case of ARGB with only the foreground color visible in the final blend. Table 55-1948 describes the blend options for transparency mode.

Table 55-1948. Blend options for transparency mode

Case	BB	AB[1:0]	Mode	Function
1	0	00	Transparency	No blending, underlying pixels are obscured
2	1	00	Transparency	Reserved
3	0	01	Transparency	The value in TRANS becomes the alpha channel of all pixels on the layer
4	1	01	Transparency	The value in TRANS becomes the alpha channel of the selected pixels on the layer
5	0	10	Transparency	Same as case 3
6	1	10	Transparency	Background color is ignored, selected pixels are completely removed, the value in TRANS is multiplied with the graphic data value (alpha) and the resultant alpha is used to blend the non-selected pixels on the layer
7	0	11	Transparency	Reserved
8	1	11	Transparency	Reserved

Figure 55-1942 –Figure 55-1945 illustrate the effect of the cases identified in Table 55-1948. In all cases there is a single active transparency layer and a white background color.

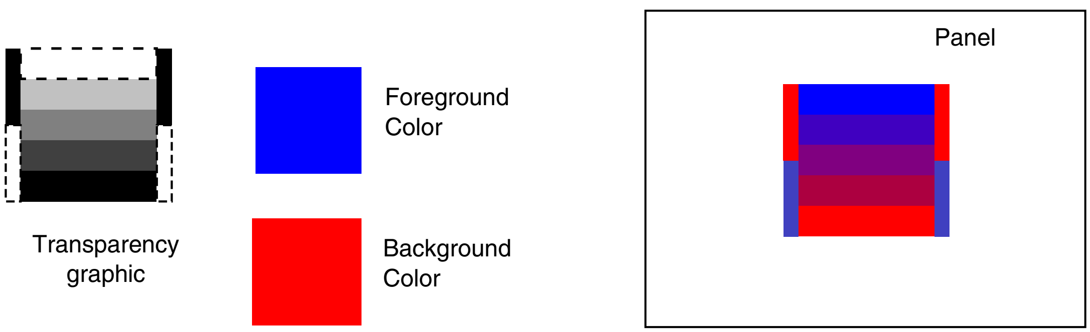


Figure 55-1942. Case 1 example (no blend)

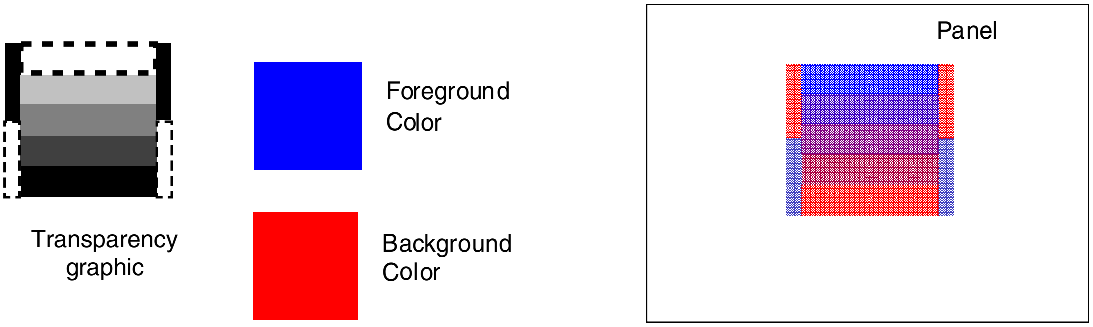


Figure 55-1943. Case 3 example (all pixels transparent)

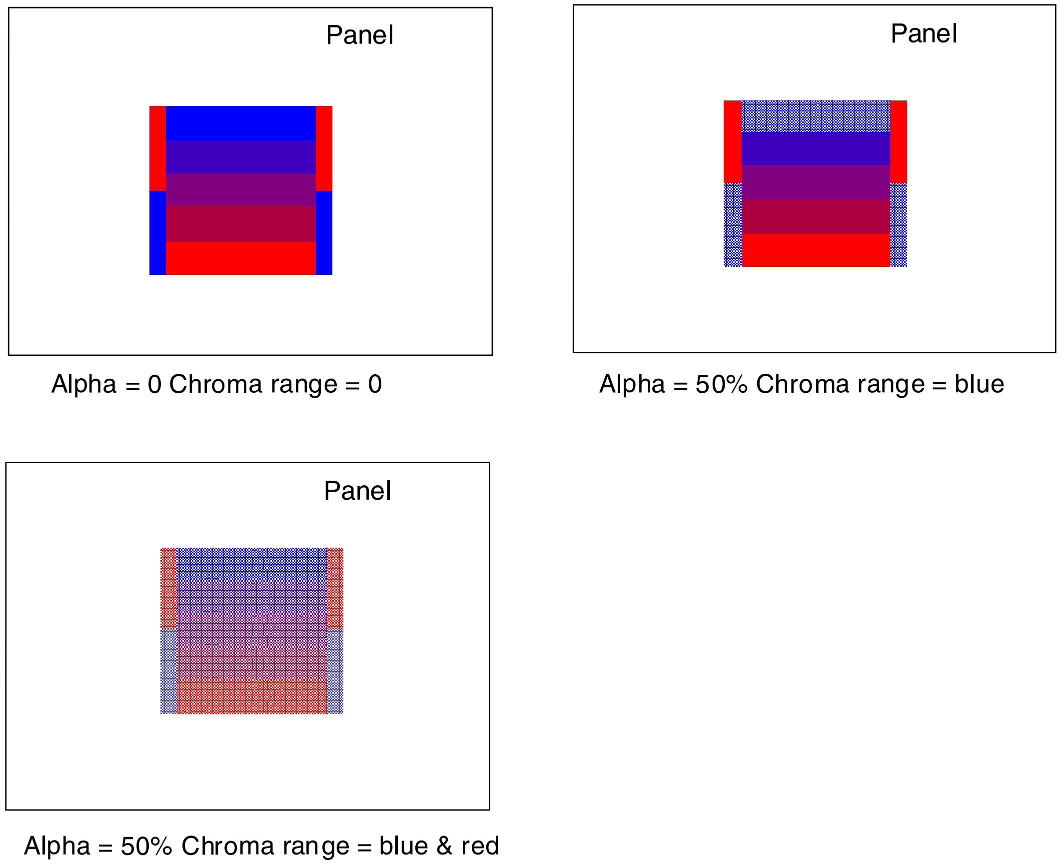
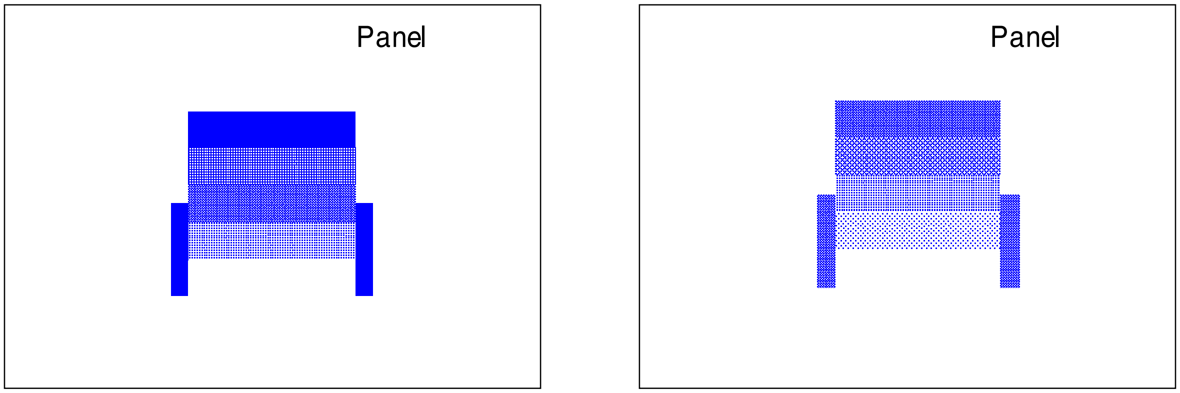


Figure 55-1944. Case 4 example (selected pixels transparent)



Alpha = 0 Chroma range = 0 Alpha = 50% Chroma range = 0

Figure 55-1945. Case 6 example (only foreground color blended)

55.6.4.8 Luminance mode

Luminance mode is a special case for the graphic data format and is defined by the BPP bit field in register 4 in the control descriptor for the layer (CTRLDESCLn_4, where n is the layer number). This value also influences the range of values for the width of the layer (see Layer size and positioning"). By choosing an appropriate format, it is possible to optimize the memory required by the graphics in use.

In luminance mode, the data in the source graphic does not contain any direct or indexed color information or alpha information. The data values in a layer in luminance mode modify the values of the pixels on underlying layers only. There are two luminance modes available: 4 bpp and 8 bpp. In both cases, the data values behave as signed integers that are added to each component of the underlying pixel. The 4 bpp mode is left-shifted with the 4 bits concatenated to form a signed 8 bpp integer. The results of the addition are prevented from overflowing, so that any result greater than 0xFF is set to 0xFF and any result less than 0x00 is set to 0x00.

The result of a blend with a luminance layer is that the intensity of the underlying pixel's color will be increased or decreased. In this way, luminance mode can be used to highlight or dim pixels on the panel without having to modify the source graphic data. Table 55-1949 describes the effect of luminance blends on an underlying pixel.

Table 55-1949. Example of a blend with a luminance mode layer

Pixel value	Luminance value	Resultant pixel
0xFF8040	0x40	0xFFC080
0xFF8040	0xC0	0x3F0000

55.6.4.9 Tile mode

Tile mode is a special case for the layer and is enabled by the TILE_EN bit field in register 4 in the control descriptor for the layer (CTRLDESCLn_4, where n is the layer number). Tile mode is only available on layers 0 to 7 and 56 to 63.

In this mode the layer size register (CTRLDESCLn_1, where n is the layer number) defines the size of the layer; however, the size of the graphic is defined in control register 7 (CTRLDESCLn_7, where n is the 64layer number). The size of the graphic must be less than or equal to the size of the layer. When tiled mode is enabled, the graphic is repeated horizontally and vertically until it fills the whole layer. The horizontal size of the tile is defined by the TILE_HOR_SIZE bit field and is restricted to be a multiple of 16 pixels. The vertical size of the tile is defined by the TILE_VER_SIZE bit field.

The graphic data for the Tile Mode can be fetched either from the system memory or from the internal CLUT/Tile memory. This is defined by the DATA_SEL bit field in register 4 in the control descriptor for the layer. If the graphic is fetched from CLUT/tile memory then it must be in RGB888 format. Otherwise the graphic can be in any previously described data format. See Figure 55-1946 for an example of a layer in tile mode.

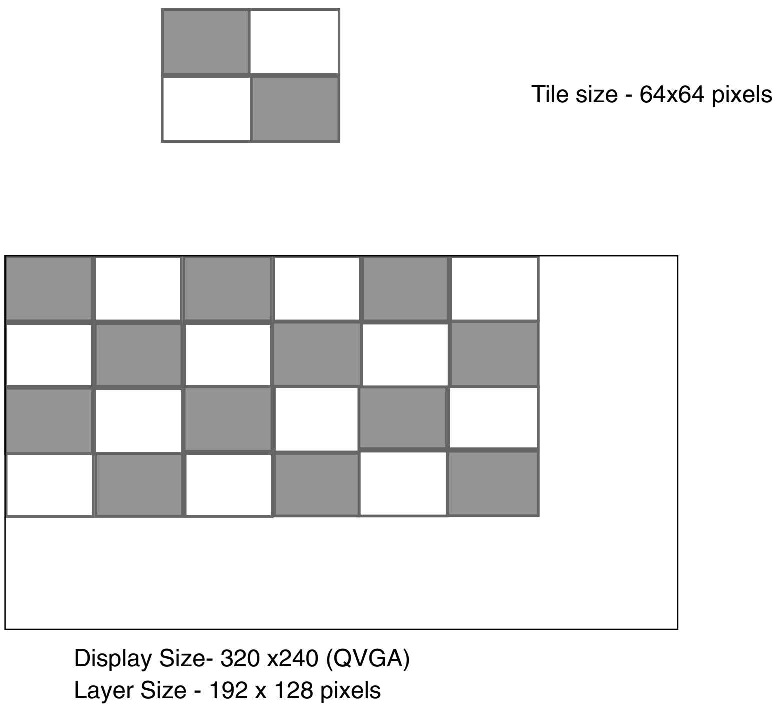


Figure 55-1946. Tile Mode

55.6.5 Hardware cursor

In addition to the 64 layers, the DCU4 also provides a special layer intended for use as a cursor. This cursor operates in 1 bpp mode and includes its own RAM area to store the graphic. The cursor may be placed at any location on the panel and includes an automatic blink option. The hardware cursor is configured using a dedicated control descriptor.

The size of the cursor is defined by register 1 in the control descriptor for the cursor (CTRLDESCCURSOR_1). The register contains two bit fields, HEIGHT and WIDTH, which determine the size and shape of the layer. Both fields are expressed in terms of the number of pixels in each dimension. The HEIGHT is limited to a maximum of 256 pixels, and the total number of pixels cannot exceed the number of bits in the cursor RAM (8192 bits).

Bits in the cursor RAM that are 0 become transparent on the panel. Bits that are 1 become fully opaque in the color defined in register 3 in the control descriptor for the cursor (CTRLDESCCURSOR_3). The DEFAULT_CURSOR_COLOR bit field is in RGB888 format.

There are restrictions on the arrangement of bits in the cursor RAM depending on how the HEIGHT and WIDTH bit fields are configured.

- When the cursor width is an integer multiple of 32 bits, the pixels in each row roll from one word in the RAM to the next one. The rightmost bit in the first word in the RAM is the t op leftmost pixel on the display. The leftmost bit in the word represents a pixel that is adjacent to the rightmost bit in the next word (in the same row). The leftmost pixel on the next row is the rightmost bit in the first word after n words that describe the first row.
- When the width of the cursor is greater than 32 bits but not an integer multiple of 32, the pixels in each row roll from one word into the next one such that the rightmost bit in the first word of the row is the leftmost bit on the display. In the final word of the row there are unused bits.

The position of the cursor on the panel is defined by register 2 in the control descriptor for the cursor (CTRLDESCCURSOR _2). The register contains two bit fields, POSY and POSX, which determine the location of the upper left pixel of the cursor in the x and y axes. Both fields are expressed in terms of the number of pixels in each axis. Placing the cursor beyond the panel area is not allowed.

The cursor can be configured to blink at a particular rate when it is enabled. The EN_BLINK, HWC_BLINK_ON, and HWC_BLINK_OFF bit fields define the blink behavior. These are in register 4 in the control descriptor for the cursor (CTRLDESCCURSOR_4). EN_BLINK enables blinking. The blinking time is based on the frame rate, and the on and off times are independently configurable. HWC_BLINK_ON configures the number of frame refresh cycles for which the cursor is

visible. HWC_BLINK_OFF configures the number of frame refresh cycles for which the cursor is not visible. For a frame refresh rate of 64 Hz, the HWC_BLINK_ON and HWC_BLINK_OFF counters give a range of on/off times up to 4 seconds.

The cursor is enabled by setting the CUR_EN bit field in register 3 in the control descriptor for the cursor (CTRLDESCCURSOR_3).

If the DCU4 detects an invalid configuration in the cursor control descriptor, then the cursor configuration is invalid and it cannot be made visible. In addition, the error flag HWC_ERR is set in the layer parameter error register (PARR_ERR_STATUS3).

The cursor RAM may be written at any time when the TFT LCD panel is not being driven with data. This means that the RAM can be modified when the DCU4 is not enabled and during the vertical blanking period.

55.6.6 CLUT/Tile RAM

The internal tile memory and color look up table (CLUT) memory share a common block of RAM internal to the DCU4. Color information in this RAM is always stored as aligned 32-bit words where the most-significant byte is the alpha component, the next byte contains the red component, the next the green component and the least significant byte the blue component (0xAARRGGBB).

This memory block can be used to store either color look-up tables or graphics for use as a tile on a layer. The content of the RAM at a specific address is defined by the control descriptor of a layer. The LUOFFS bit field in the layer control descriptor defines the starting address of the area, and the BPP and TILE_EN bit fields define what type of use the RAM area has.

In Figure 55-1947 three areas of the RAM are defined for different purposes. Area A is used by layer 1 as a CLUT for its 4 bpp graphic. Area B is use by layer 5 as a store for its tile graphic. Area C is used by layers 2, 7, and 9 as a CLUT for their 8 bpp graphics.

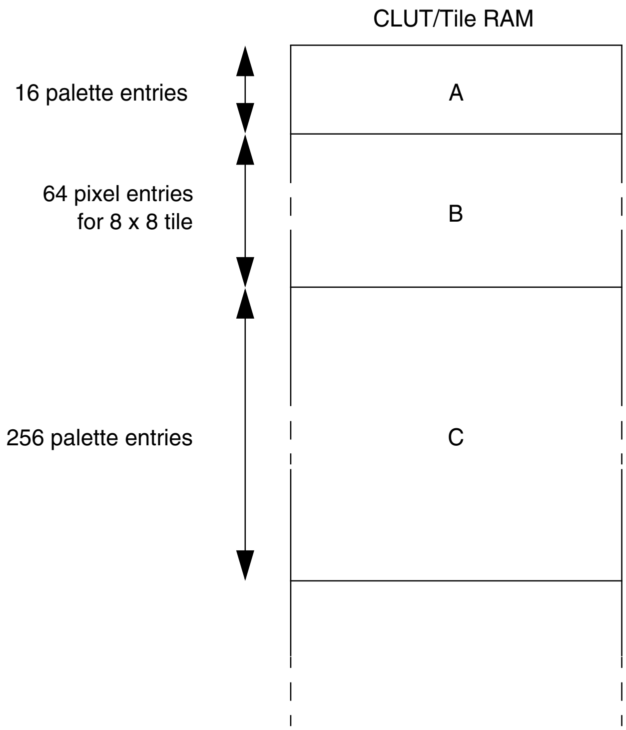


Figure 55-1947. An example of use for the CLUT/Tile RAM

The CLUT/Tile RAM is mapped in the DCU4 32K memory space from address 0x2000 to 0x3FFF. This gives 2048 entries, which provides up to eight full CLUTs for 8 bpp layers.

The CLUT/Tile RAM may be written at any time when the TFT LCD panel is not being driven with data. This means that the RAM can be modified when the DCU4 is not enabled and during the vertical blanking period.

55.6.7 Gamma correction

The gamma table allows the user to define an arbitrary transfer function at the output of each color component. The function (Gamma correction) is applied to each pixel after all blending is complete and before the data is driven to the TFT LCD panel. Gamma correction is optional and can be used to adjust the color output values to match the gamut of a particular TFT LCD panel, or to perform data inversion or data length reduction on each component.

$$\mathrm{output\_color\_component} = \mathrm{gamma\_table}[\mathrm{input\_color\_component}]$$

The table is arranged as three separate memory blocks within the DCU4 memory map; one for each of the three color components. Each memory block has one entry for every possible 8-bit value and the entries are stored at 32-bit aligned addresses. This means that the upper 24 bits are not used while reading/writing the gamma memories. See Figure 55-1948 for details of the memory arrangement.

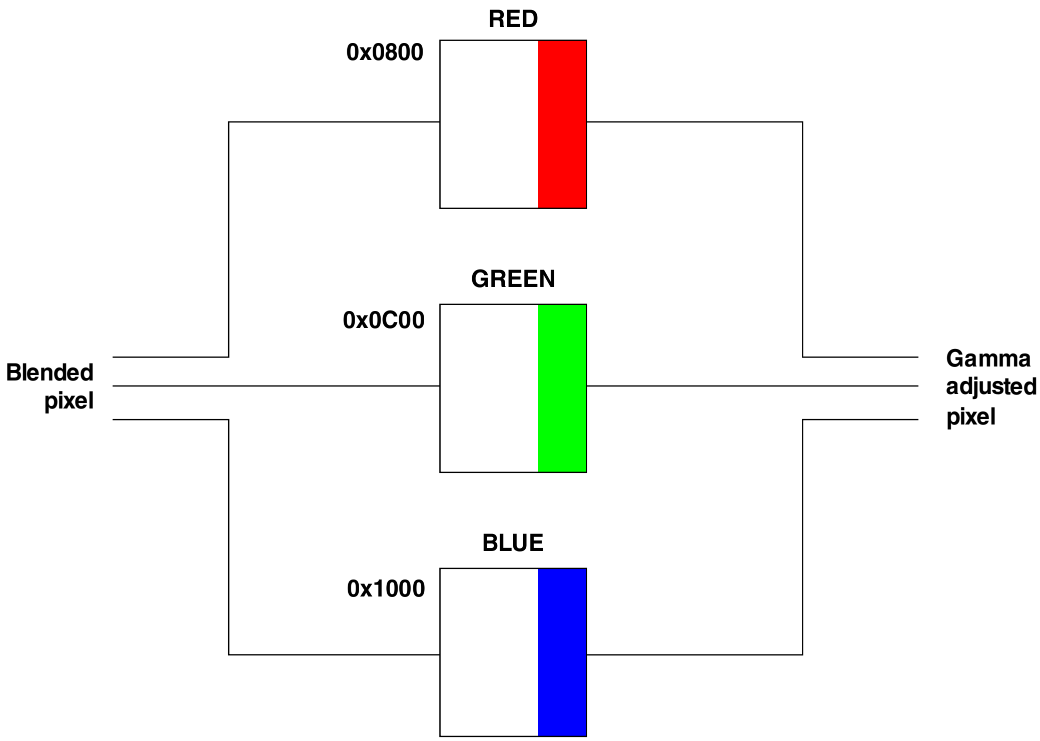


Figure 55-1948. Gamma Correction Table Organization

The gamma table can only be read or written when the DCU4 is not enabled or during the vertical blanking period.

55.6.8 Temporal Dithering

This is a technique that allows the emulation of a color resolution higher than the resolution supported by the display. It is done by changing the intensity values over time sent to the display. The averaging done by the human eye gives the impression of the intensity of such alternating pixels as an interim value between the two supported intensity values. Temporal dithering is enabled by the DCU_MODE[DITHER_EN] bit.

The key features of the dithering block are:

- Temporal dithering increases the optically perceived depth of a limited TFT display.
- Supports display with 5-8 bits resolution per color component.
- Independent dither control parameters per color component.
- Support for safety mode.

Temporal dithering is enabled and controlled by the EN_DITHER, ADDB, ADDG, and ADDR bits in the DCU_MODE register. The ADDx fields are each 2 bits wide and select how many bits to add to each color component. The typical setting is 8 minus the number of bits in each component required by the display.

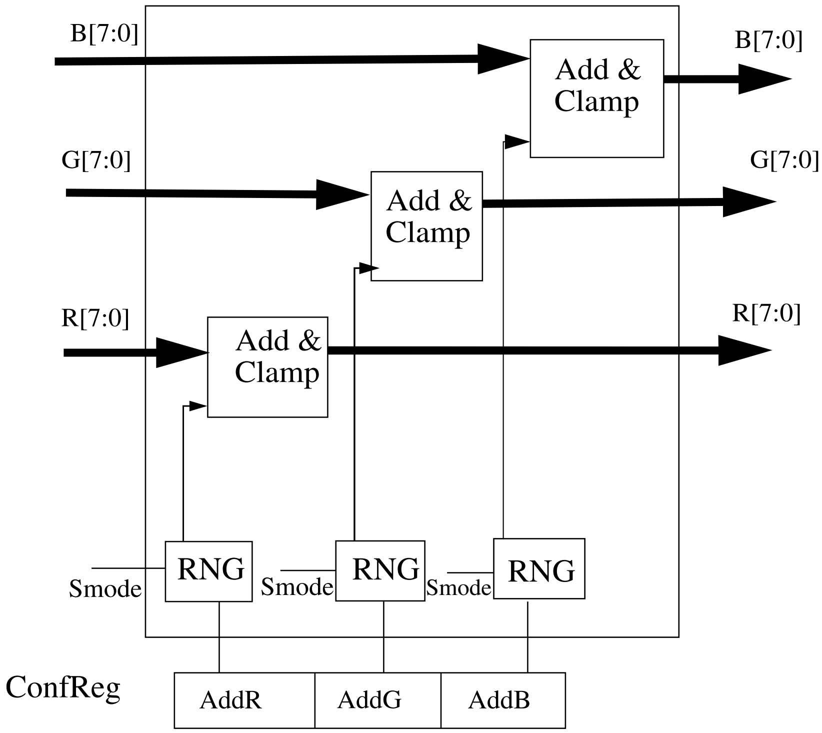


Figure 55-1949. Dithering block diagram

The Random Number Generator (RNG) provides a random number of up to 3 bits. The number of bits provided is selected by the values of each component's ADDx field.

When pixels from a safety layer are encountered the RNG output is forced to 0 which effectively disables the temporal dithering block and these pi xels are passed to the display unmodified.

The Add & Clamp block adds the eight bit pixel value to the thr ee bit number generated by the RNG. The result is then clamped to the range of 0..255.

55.6.9 Special DDR Mode

Special DDR mode is a special configuration that optimizes the use of an SDRAM memory by the DCU4 by forcing the DCU4 to fetch data in optimal chunks. In this special mode only the six highest priority layers (0:5) are available in the DCU4. This mode is enabled using the DCU_MODE[DDR_MODE] bit.

When this mode is enabled the DCU3 will fetch data in 32-byte chunks if the layer is encoded in 8, 16 or 32 bpp formats, thus optimizing the SDRAM throughput. Any layers in 1, 2, 4 or 24 bpp formats will be fetched using normal access thus they will not benefit from any optimization and may disrupt optimal access for any 8-, 16- and 32-bit formatted layers if both are in the SDRAM. Therefore it is highly recommended to store any 1, 2, 4 or 24 bpp layers in non-SDRAM memory such as on-chip SRAM.

Depending on the layer configuration in use this mode may also benefit other synchronous memory interfaces such as QuadSPI.

This mode only permits operation in four-layer blend mode, therefore, the DCU_MODE[BLEND_ITER] field must be set to 4.

55.6.10 Run Length Encoding (RLE) Mode

RLE mode is available on the all the layers and allows the DCU5 to load RLE compressed data from memory and directly decode it for use on the panel. The mode is enabled using the RLE_EN bit in the control descriptor 4 register CTRLDESCLn_4. This mode is only available when Special DDR mode is enabled and can only be used on layers equal to a number of rle layers at a time . Maximum value of number of rle layers can be 6. In addition the mode only supports 8 bpp, 16 bpp (RGB565, ARGB1555, ARGB4444, APAL8), 24 bpp and 32 bpp BGRA8888 formats.

When enabled, the DCU5 fetches encoded data on start of arbitration and provides the decompressed data according to the new arbitration scheme to the normal DCU blend process.

The following figure shows the dataflow for the RLE decoded data.

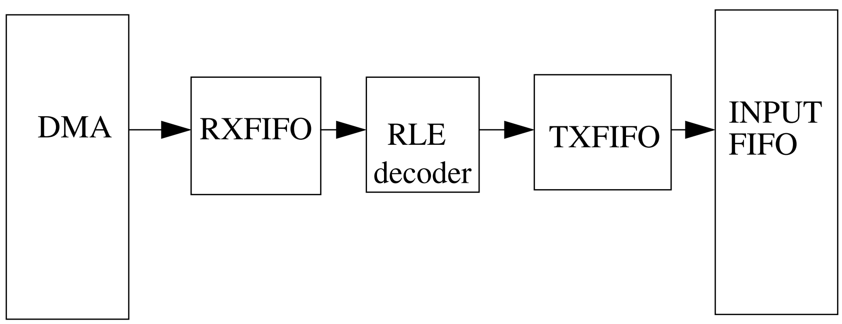


Figure 55-1950. RLE Decoding in DCU5

The decoded data is read by the input FIFO once at l east 8 bytes are available in the TXFIFO. The size of RXFIFO is 64x8 bits while the size of TXFIFO is 16x8 bits.

If more than the number of rle layers have RLE_EN set, then the error flag RLE_ERR is asserted.

55.6.10.1 RLE Decoding Scheme

Before enabling an RLE encoded layer configure the COMP_IMSIZE register with the size of the compressed image. The decoder expects to read COMP_IMSIZE bytes from the image and produce from that the number of pixels specified in the layer Control Descriptor 11 register.

The format of RLE encoded layers is as follows:

- The first data byte read at the image address (Layer Control Descriptor 3) is a command byte (CMD[7:0]).
- The ms bit (CMD[7]) indicates if the following bytes are raw or compressed pixels. One pixel can be 8-bit, 16-bit, 24-bit or 32-bit wide, depending on the BPP bit field in the Layer Control Descriptor 4 register.
- The remaining 7 command bits (CMD[6:0]) specify the number of raw or compressed pixels that follow the command byte. This count is offset by 1 such that a value of 0 means one pixel follows.
- For compressed pixels (CMD[7] = 1), only one pixel follows the command byte. This pixel is repeated count+1 times on the layer. The pixel size may be 8, 16, 24 and 32 bits.
- For raw pixels (CMD[7] = 0) count + 1 pixels follow the command byte and these are included on the layer as is. The pixel size may be 8, 16, 24 and 32 bits.
- If there is more data to decode, a new command follows after CMD+(1*{Pixel width}) bytes. This encoding is repeated until the whole image is decoded.

55.7 Timing, Error and Interrupt Management

The DCU4 can detect and raise status and error flags when the status of the system changes and when configuration or operational errors are detected.

55.7.1 Synchronizing to panel frame rate

Since the DCU4 fetches data directly from memory independently of the CPU, there is the possibility that changes to the DCU4 layer configuration or content can create incoherent content on the panel. To help avoid this situation there are five timing control flags that define when the DCU4 recognizes and locks changes to its configuration. These can be used to manage changes to control descriptors, CLUT or tile memory, or source graphics and so avoid coherency problems on the panel. All the timing flags are in the INT_STATUS register and can be used to generate interrupts from the DCU4.

The VS_BLANK and LS_BF_VS flags give indication of the start of the vertical blanking period. The VS_BLANK flag is set at the beginning of the vertical blanking period. The LS_BF_VS flag is set a given number of horizontal lines before the start of the vertical blanking period; the given number of lines is defined by the LS_BF_VS bit field in the THRESHOLD register.

The PROG_END flag indicates that the DCU4 has begun a new panel refresh period. Further changes to the layer control descriptors should not occur until the transfer is complete.

The DMA_TRANS_FINISH flag indicates that the DCU4 has completed fetching all data from memory in the current panel refresh cycle. This normally precedes the vertical blanking period and indicates that it is possible to change the contents of a memory that contains graphics used by the DCU4.

The VSYNC flag indicates that the DCU4 has begun the next panel refresh period.

An additional flag called LYR_TRANS_FINISH is available to indicate when the DCU4 has completed transferring the layer configuration for this refresh cycle. This operates as a functional interrupt rather than a timing interrupt but can also help to ensure data coherency.

55.7.2 Managing the DCU4 FIFOs and DMA activity

The DCU4 fetches graphic data directly from internal and external memory using a dedicated DMA system and manages the output of data to the TFT LCD panel such that the panel always receives the pixel information when expected. Since the panel is sharing access to memory with the system DMA and CPU it cannot depend on the required data always being available at all times. It therefore it uses input and output FIFOs to temporarily store incoming and outgoing data until required and thus reduces the opportunity for the panel to be starved of pixel data.

The DCU4 manages the supply of graphic data to its format conversion and blending stages using input FIFO buffers that are 256 x 64 bits in size. The data that is driven to panel is managed using an output FIFO buffer that is 128 pixels in size. See Figure 55-1 for a diagram of the input FIFO and output FIFO operation in the DCU4

The input FIFOs are not accessible to the user but it is possible to set thresholds that control the DCU4 behavior when the FIFOs are becoming full or empty and observe when the lower and higher thresholds are reached. The DCU4 also provides information on those conditions where data is expected from an input FIFO but is not available. This can help detect and avert situations where the DCU4 is running out of data to send to the panel.

The FIFO thresholds are set in the THRESHOLD_INPUT_BUF_1/2/3 registers. The upper thresholds are set by the INP_BUF_Pm_HI bit fields (where m is the position of the pixel in the blend stack) and these set the point at which the DCU pauses fetching data from memory. The maximum size of any DMA burst is fixed to 16 pixels and so is dependent on the graphic encoding. The lower thresholds are set by the INP_BUF_Pm_LO bit fields.

Each of the six input FIFOs has two flags that indicate whether the FIFO has reached its upper or lower threshold. The Pm_FIFO_HI_FLAG flags (where m is the position of the pixel in the blend stack) indicates that the input FIFO has reached the upper threshold. The Pm_FIFO_LO_FLAG indicates that the input FIFO has less data than its low threshold. Depending on when the low threshold is reached this may indicate a number of scenarios

- The expected graphical data is not available for the DCU4 to load
- The DCU4 is reaching the end of a frame and does not need to load any more data
- The blend stack does not need pixels of this priority

In the situation where the data is not available to the DCU4 then there may or may not be an impact to the data visible on the panel. The DCU4 provides additional information in this case via the Pm_EMPTY flags in the INT_STATUS register and the LINE and PIXEL values in the UNDERRUN register. The Pm_EMPTY flags indicate which position in the blend stack that the latest problem occurred and the LINE and PIXEL entries indicate approximately where on the panel the condition occurred. Taken together this information should allow identification of the source graphic/layer which was too slow to be loaded.

In the situation where the output FIFO is full then it is possible for the DCU4 to accept a delay before it requires to use the incoming data.

The output FIFO is not accessible to the user but it is possible to set thresholds that control the DCU4 behavior when the FIFO is becoming full or empty and observe the lower threshold. This can help detect and avert situations where the DCU4 is running out of data to send to the panel.

The buffer thresholds are set in the THRESHOLD register. The upper threshold is set by the OUT_BUF_HIGH bit field and this indicates that sufficient data exists in the output buffer and processing should stop until the DCU4 uses some of the values in the FIFO. If this value is set too low then the possibility of the DCU4 running out of data to drive the panel is increased. The lower threshold is set by the OUT_BUF_LOW bit field.

When the output FIFO has emptied below its low threshold (OUT_BUF_LOW bit field) it sets the UNDRUN bit. In an under run situation there may or may not be an impact to the data visible on the panel. The impact depends on whether the DCU4 is reaching the end of a frame and how close to running out the threshold is set.

The best guide to indicate whether the DCU4 is able to supply the required pixel information to the panel is the output buffer. If the output is indicating that it is running out of data, the input FIFOs and UNDERRUN register may help identify the areas of memory that are restricting the supply of data. Using these indicators can help to set the DCU4 thresholds and ensure that the data throughput on the MCU is balanced correctly for all master devices.

Finally, note that the number of DCU4 clock cycles to fetch and blend each pixel increases with the depth of the blend stack. However, the time taken to process the pixel data is fixed by the timing requirements of the panel. Therefore, for full performance across all color encodings the ratio between the DCU4 clock and the pixel clock must increase as the blend stack depth increases:

- For two pixel blend, maximum supported pixel clock is DCU4 clock/2.
- For three pixel blend, maximum supported pixel clock is DCU4 clock/3.
- For four pixel blend, maximum supported pixel clock is DCU4 clock/4.
- For five pixel blend, maximum supported pixel clock is DCU4 clock/5.
- For six pixel blend, maximum supported pixel clock is DCU4 clock/6.

55.7.3 Error detection

The DCU4 asserts error flags when errors are detected in its configuration or when the user attempts to modify the configuration at an invalid point in the panel refresh period or when it is unable to access the required source data. The error flags may raise an interrupt if enabled to do so by the related mask bit in the corresponding mask register.

Error flags are stored in the PARR_ERR_STATUS1, PARR_ERR_STATUS2, PARR_ERR_STATUS3, and INT_STATUS registers.

Errors in the DCU4 layer configuration are collected in the PARR_ERR_STATUS1 and PARR_ERR_STATUS2 registers.

The flags Ln (where n is the layer number) indicate an error in the configuration of the layer which can be either an invalid tile mode size or a layer with a horizontal dimension that is smaller than the minimum size defined by the layer encoding (see Layer size and positioning").

Reads of CLUT/Tile RAM during the period when the TFT LCD panel is being updated do not return the CLUT/Tile RAM content.

Errors caused when the DCU is unable to access its required source data are collected in the INT_STATUS register. These errors are indicated by the UNDRUN flag, the Pm_EMPTY flags, and the Pm_FIFO_LO_FLAG flags (where m is the position in the blend stack)

55.7.4 Interrupt generation

The DCU4 generates interrupt through four lines that are controlled by the contents of ten registers:

- INT_STATUS
- INT_MASK
- PDI_STATUS
- PMASK_PDI_STATUS
- PARR_ERR STATUS1
- PARR_ERR STATUS2
- PARR_ERR STATUS3
- MASK_PARR_ERR STATUS1
- MASK_PARR_ERR STATUS2
- MASK_PARR_ERR STATUS3

There are four interrupt status lines defined.These lines are grouped as follows

- Timing based interrupts:
    - VSYNC
    - LS_BF_VS
    - VS_BLANK
    - PROG_END
    - DMA_TRANS_FINISH
- Functional interrupts:
    - UNDRUN
    - LYR_TRANS_FINISH
    - CRC_READY
    - CRC_OVERFLOW

  - P1_FIFO_HI_FLAG
  - P1_FIFO_LOW_FLAG
  - P1_EMPTY
  - P2_FIFO_HI_FLAG
  - P2_FIFO_LOW_FLAG
  - P2_EMPTY
  - P3_FIFO_HI_FLAG
  - P3_FIFO_LOW_FLAG
  - P3_EMPTY
  - P4_FIFO_HI_FLAG
  - P4_FIFO_LOW_FLAG
  - P4_EMPTY
  - P5_FIFO_HI_FLAG
  - P5_FIFO_LOW_FLAG
  - P5_EMPTY
  - P6_FIFO_HI_FLAG
  - P6_FIFO_LOW_FLAG
  - P6_EMPTY
  - IPM_ERROR
- Parameter error interrupts
  - Layer Error
  - Signature Calculator Error
  - Display Error
  - HWC_error
- PDI-related interrupts (pdi_int)
  - This includes PDI related interrupts. See PDI-Related Interrupts for a description.

When any interrupt occurs, the host can identify which type of interrupt has occurred by reading the interrupt status register and PARR_ERR status registers.

55.8 Register protection

There is a customized register protection scheme on the DCU4. The scheme provides a mechanism to protect certain registers in the DCU4 from being written.

55.8.1 Operation of scheme

The register protection scheme provides a two-step protection scheme for the protected register.

Firstly, each register has an associated soft lock bit (SLB) that prevents further writes to the register when it is set. Each SLB has a corresponding write enable (WEN) bit that must be set in the same write operation as the SLB. The SLB can be set or cleared by writing a '1' or '0' to it while its WEN bit is set. The SLB bits are in the Soft Lock Registers L0 and L1, DISP_SIZE, HSYNC/VSYNC_PARA, POL, L0_TRANSP and L1_TRANSP registers.

Secondly, there is a hard lock bit (HLB) in the Global Protection Register which prevents all changes to soft lock bits. The HLB can only be cleared by a system reset.

If a write is made to a register whose SLB is set then a transfer error occurs that generates a data abort on the CPU. Similarly if the HLB is set then any write to the SLB registers causes a transfer error.

55.8.2 List of protected registers

The register protection scheme applies to the following registers:

- All Layer 0 control descriptors CTRLDESCL0_1 to CTRLDESCL0_9
- All Layer 1 control descriptors CTRLDESCL1_1 to CTRLDESCL1_9
- DISP_SIZE
- HSYNC_PARA
- VSYNC_PARA
- SYNPOL

55.9 Safety Mode

Safety layers are used in a multi-layer DCU4 environment for the purpose of guaranteeing that the content is driven to the display regardless of the setting of remaining layers and the pixel manipulation algorithms of the DCU4. Features such as this are a requirement from qualification institutes to be able to reach a safety level of SIL2 or ASILB. The DCU4 has two safety layers (Layer 0 and Layer 1) which also have the highest priority. When Safety Mode is active the safety layers can use chroma keying for complex area description, however alpha blending for the layer is always ignored. Additionally, if a layer has safety mode enabled then a layer format of 32 bpp or luminance is not allowed. Using these formats causes the layer to be disabled.

Safety Mode is implemented using a signature calculator module implemented inside the DCU4 that calculates two signatures (pixel value and pixel position) for a predefined area of the frame. The user makes layer 0 and/or layer 1 active as a safety layer, defines the window/area of the pixels for which the signature is to be calculated, and enables safety mode. When enabled, the signature calculator starts to calculate the signature after the first pixel in the selected area is available and after the start of the next frame (VSYNC). It is also possible to calculate the signature value for all pixels if DCU_MODE[TAG_EN] = 0.

As the pixels in the selected area become available they are "tagged" by the DCU4, except for those removed by chroma-keying. These tags identify the pixels to be included in the signature calculation. The signature calculation itself is an industry-standard CRC.

The DCU4 asserts the CRC_READY flag at the end of any frame which has Safety Mode enabled. This can be used to indicate the completed signature calculations for each full frame of pixels after the mode is enabled. The completed signature can then be compared against a pre-calculated value with any difference indicating that the pixels displayed did not match what was expected. The signature calculator then continues to calculate the CRC for the next frame. If the CRC_READY flag is not processed within one frame time period, then the CRC_OVERFLOW interrupt is issued and the latest calculation overwrites the previous value. Since the CRC_READY flag is set at the end of any frame with Safety Mode enabled, it is possible that a full frame has not yet been completed and therefore no signature calculation exists for the frame that set the flag.

If the user has set the NEG bit for the DCU4 which indicates that the pixels fed to the display are inverted, then the value CRC is calculated on non-inverted values. The position CRC, however, remains as is.

Normal arbitration takes place only when a pixel has content on layer 0 and layer 1 but where Safety Mode is enabled in layer 1.

The polynomial used for CRC calculation is (x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1).

For the value CRC, the 24-bit value of each output pixel (after decoding) is sent to the polynomial. For the position CRC, the value sent is (pixel_delta_y * display.delta_x) + (pixel_delta_x + 1).

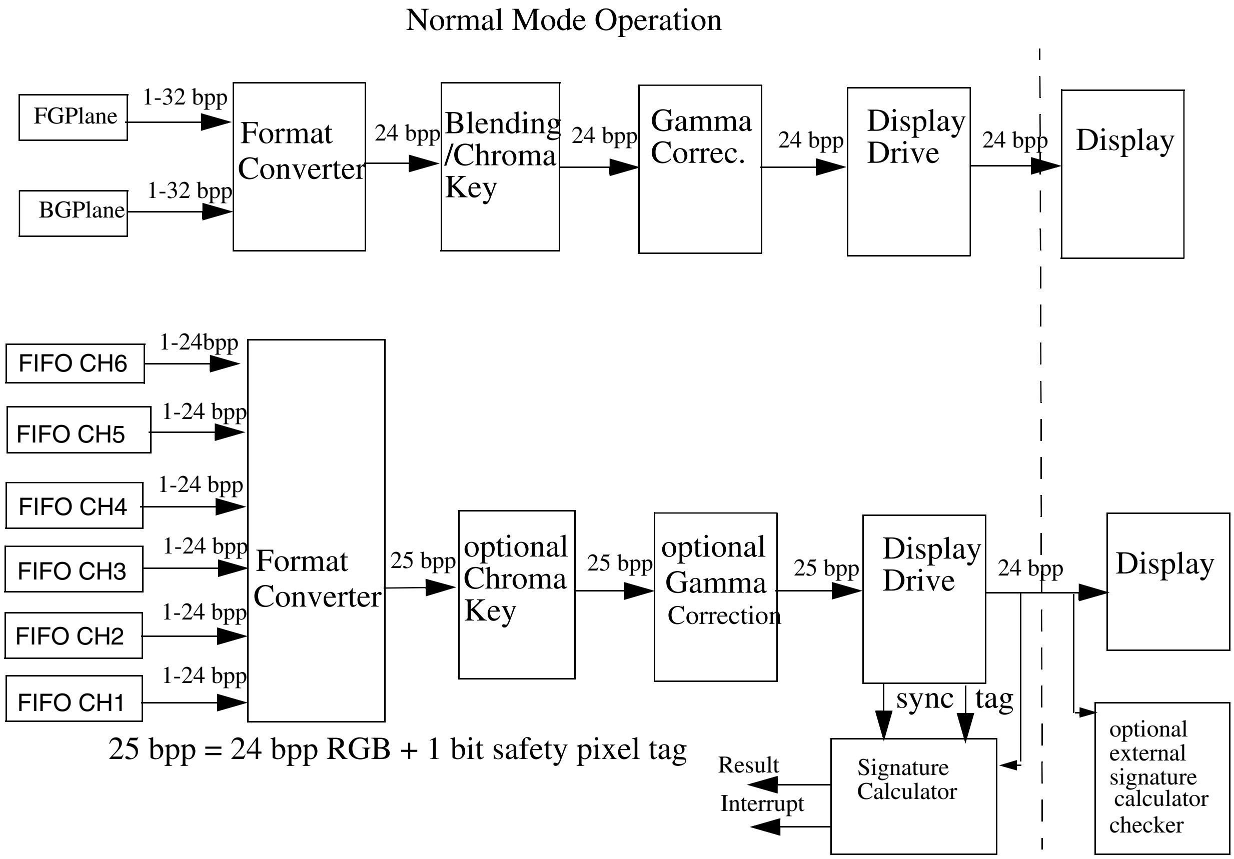


Figure 55-1951. Safety Mode Block Diagram

55.9.1 CRC Area Description

55.9.1.1 Relationship between various input signals

To configure the CRC calculation:

1. CRC_VAL and CRC_POS are calculated when Safety Mode is enabled using DCU_MODE[SIG_EN].

2. The CRC can be calculated for part of the panel as shown in the following figure. The green portion on the panel is identified using the SIGN_CALC_1 and SIGN_CALC_2 registers which defined the size of the area and the location of the area respectively. If SIGN_CALC_1 and SIGN_CALC_2 are configured appropriately this calculation will cover the whole panel.

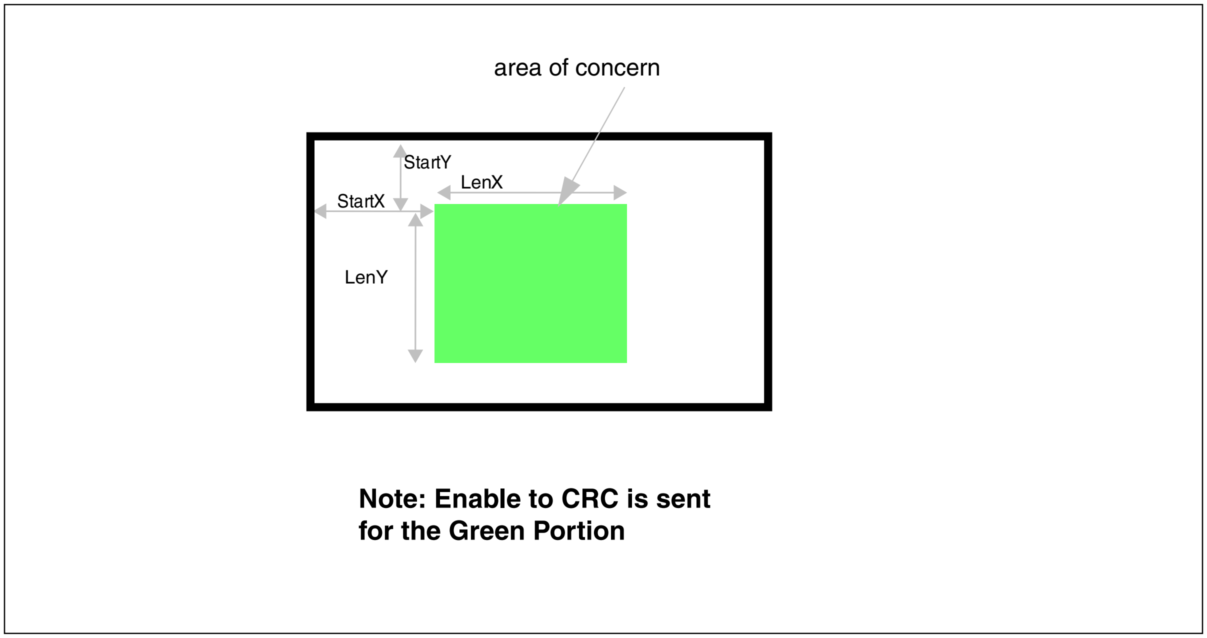


Figure 55-1952. Safety Mode enabled for part of the screen

3. The CRC can be calculated exclusively for layers 0 and 1 by setting DCU_MODE[TAG_EN] and enabling the SAFETY_EN bit in control descriptor 4 of each of the layers. In this configuration the CRC is calculated using values from the layers only where they intersect the area of interest defined in SIGN_CALC_1 and SIG_CALC_2. An example is shown (in dark pink) in the following figure.

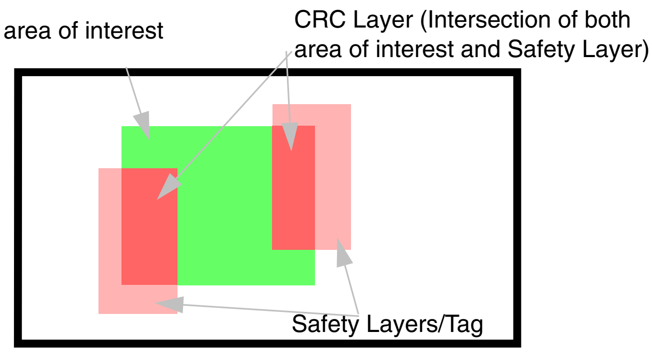


Figure 55-1953. Safety Mode with tag bit high

4. Layer 0 and 1 (i.e., safety layer) in safety mode does not supports blending or luminance offset.

55.9.1.2 Features

- SC support area modes mentioned in Table 55-1950
- Supports calculation of CRC on the pixel value only
- CRC is calculated with the initial value as 32'h000000000
- CRC does not support:
  - Any modification of input bytes
  - Any modification of the output CRC value before reporting.
- When PDI is enabled with the Layer 0 and 1 (safety enabled) in a single sector, then PDI would act as BG layer and Layer 0 as a FG Layer.CRC would be calculated over a single L0 Layer.

Table 55-1950. Supported Area

Area	Tag Value	Note
Full	1'b0	StartX = 0 StartY = 0 LenX = Screen Size LenY = Screen Height
Part	1'b0	All the parameter have value other the one mentioned above as shown in Figure 55-1952.
Safety Layer (Layer 0 and 1 only)	1'b1	Part of the safety Layer would depend on • Part lying within the area of concern defined by the StartX startY LenX LenY • Part deleted by the chroma keying functionality

55.9.1.3 Summary of Operation

The area included in the CRC calculation is summarized in the following table. The initial value on all CRC calculations is 0x00000000.

Table 55-1951. CRC calculation area

Area	DCU_MODE[TAG_EN]	Note
Full	0	SIGN_CALC_2[SIG_HOR_POS] = 0 SIGN_CALC_2[SIG_VER_POS] = 0 SIGN_CALC_1[SIG_HOR_POS] = Panel Width SIGN_CALC_1[SIG_VER_POS] = Panel Height
Part	0	The SIGN_CALC parameters have values other than those mentioned above. See figure 11-86.
Safety Layer (Layer 0 and 1 only)	1	The included portion of the safety layers depends on: – The portion lying in the area defined by SIGN_CALC1 and SIGN_CALC12 – The pixels removed by chroma keying functionality

55.10 Parallel Data Interface (Camera Interface)

55.10.1 PDI Interface Description

55.10.1.1 Introduction

This block extracts the timing and pixel information from an external video source and passes it to the DCU4 block to synchronize with the timing and display the pixel information on the TFT LCD screen. The BGND layer in the DCU3 is replaced by the incoming video stream.

The PDI requires that the incoming video stream match the resolution and timing at which the DCU4 is driving the TFT LCD panel and the incoming stream must not be interlaced.

The PDI can also be configured in slave mode in which case it ignores the pixel information from the external video source and only passes the timing information to the DCU4 to synchronize with. In this instance the DCU3 will continue to operate as normal (the BGND layer will not be altered) but will use the timing from the PDI.

The PDI shares configuration registers with the DCU4.

55.10.1.2 PDI Interaction With Other Modules

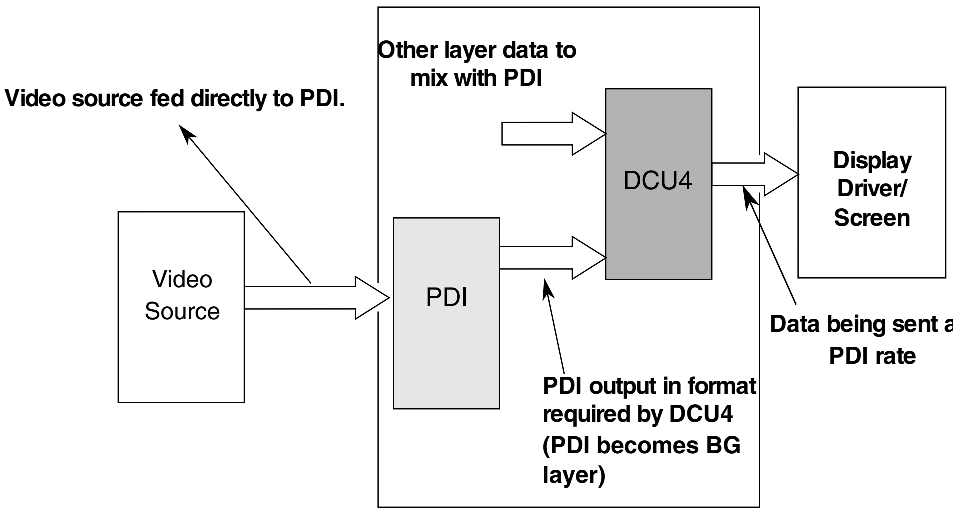


Figure 55-1954. PDI Interacting Directly with the External Sensor

In Figure 55-1954, PDI directly accepts the data from the external video source. External device must support the interface mentioned in the document.

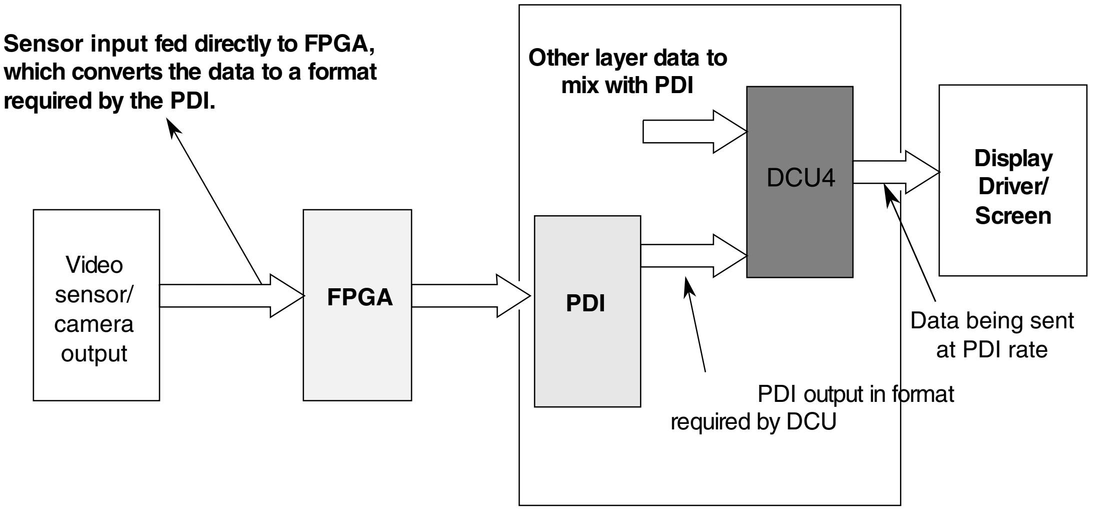


Figure 55-1955. PDI Interacting with FPGA in Between

As shown in the figure above, the video stream is sent to a decoder or an FPGA which alters the incoming stream to a format which is compatible with the PDI. A decoder/ FPGA is required if a video source with an analog output format (e.g. NTSC/PAL) is used. The decoder should perform analog-to-digital conversion on the stream and ensure the timing match that of the DCULite. The incoming stream may also need to be de-interlaced.

The PDI is compatible with various input formats:

- Normal Mode: The PDI clock frequency must be equal to the pixel clock frequency required by the TFT display driver.
- Narrow Mode: The PDI clock frequency is double the desired pixel clock frequency.
- External Synchronization: The timing signals (HSYNC, VSYNC, and DE) each have a dedicated input pin. (DE is optional).
- Internal Synchronization: The timing signals (HSYNC, VSYNC and DE) are embedded in the data stream, as such only the DATA and PCLK inputs are required. (See ITU-R BT.656 sync information extraction.)

Before the DCULite locks onto the PDI timing signals it will run on the internal DCULite clock. After lock has been achieved, the DCULite will switch to the clock from the PDI stream and the this is then used to send data and timing signal to TFT/LCD display driver.

In all cases, the resolution of the incoming stream and the HSYNC and VSYNC frequency must be the same as that for TFT screen. All the horizontal parameters (Front Porch width, Back Porch width, Pulse width) and vertical parameters (Front Porch width, Back Porch width, Pulse width) must be same as that of TFT screen.

When PDI is the background layer, no other layer can be a background layer for that particular frame. Only one background layer is possible i.e. PDI Layer when PDI is enabled.

55.10.1.3 Features

The PDI supports the following:

- RGB565, RGB666, 8 bit monochrome format,YCbCr422 mode
- Max input frequency of 32 MHz in 8/16/18 normal mode input
- Max input frequency of 64 MHz in 8 bit muxed (narrow) mode.
- External Synchronization using PDI_Hsync, PDI_Vsync, and PDI_PCLK
- External Synchronization using PDI_Hsync, PDI_Vsync, and PDI_PCLK, and PDI_DE
- Internal synchronization using PDI[17:0] and PDI_PCLK is supported for RGB565 and YCbCr422 muxed modes only.

Table 55-1952. Supported RGB formats and sync formats

RGB Format	Data Input Bus
8-bit monochrome	8 bit
RGB565	16 bit
RGB666	18 bit
RGB565 muxed (uses narrow mode)	8 bit
YCbCr	8 bit
Sync Format	**Pin Used**
Internal Sync (Valid only for RGB565 and YCbCr in narrow mode)	PCLK
External Sync	PDI_HSYNC, PDI_VSYNC, PDI_PCLK
External sync (with data En)	PDI_HSYNC, PDI_VSYNC, PDI_PCLK, PDI_DE

55.10.1.4 ITU-R BT.656 sync information extraction

According to ITU-R BT.656 recommendation, the incoming digital video will have a pdi_clk signal and 8 data bits. The data bits can contain both the video data and the timing reference signals (VSYNC and HSYNC).

The timing signals are encoded at the start and end of each line by timing reference codes known as Start of Active Video (SAV) and End of Active Video (EAV). The SAV and EAV codes are identified by their preamble of three bytes (0xFF,0x00,0x00). Due to this, neither 0x00 or 0xFF can be used during the Active Video Data. The preamble is followed by the XY status word which contains a Field Bit (F), a Vertical blanking bit (V) and Horizontal blanking bit (H) and four protection bits for single bit error correction and detection.

The H bit is set to 1 to denote an EAV — that is the end of a line, and the beginning of the horizontal blanking period. The V bit is set to 1 to denote the beginning of the vertical blanking period. The F bit is used for interlaced video to denote if the forthcoming line is odd or even.

The remaining 4 bits contains make up the protection bits for single bit error correction and detection. It should be noted that F and V fields are only allowed to change as part of EAV sequences i.e during transitions from H=0 to H=1.

An entire line of video comprises Active Video + Horizontal Blanking (from the start of the EAV code until the end of the SAV code) and Vertical Blanking (the space where V = 1).

NOTE

This device supports only 8-bit, non-interlaced, video. The Field (F) value is ignored.

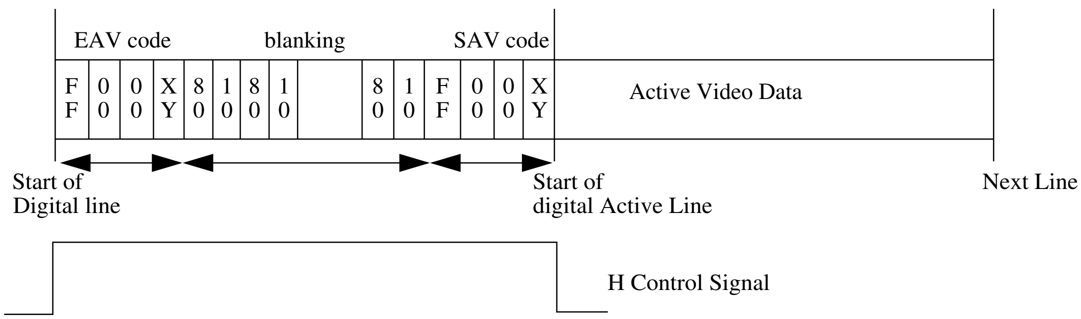


Figure 55-1956. ITU-R BT.656 8 bit parallel data format for 525 video system

Table 55-1953. Control Byte Sequence for 8-bit/10-bit video

Data Bit	FirstWord (FF)	SecondWord (00)	ThirdWord (00)	Fourth Word (XY)
D7(MSB)	1	0	0	1
D6	1	0	0	F
D5	1	0	0	V
D4	1	0	0	H
D3	1	0	0	P3
D2	1	0	0	P2

Table continues on the next page...

Table 55-1953. Control Byte Sequence for 8-bit/10-bit video (continued)

Data Bit	FirstWord (FF)	SecondWord (00)	ThirdWord (00)	Fourth Word (XY)
D1	1	0	0	P1
D0 (LSB)	1	0	0	P0

The bit definitions for the status word XY are:

F = 0 for field 0

F = 1 for field 1

V = 1 during vertical blanking period

V = 0 when not in vertical blanking

H = 0 at SAV

H = 1 at EAV

P3 = V XOR H

P2 = F XOR H

P1 = F XOR V

P0 = F XOR V XOR H

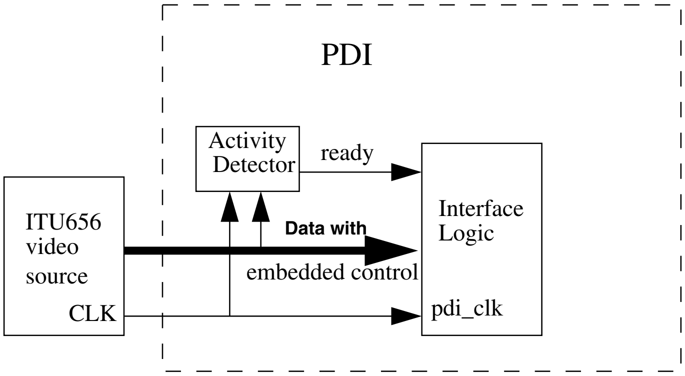


Figure 55-1957. PDI Input data mode

Figure 55-1957 represents the scenario in which data from an ITU-R BT.656 compliant video source is fed into the PDI interface. The incoming data includes codes that trigger the start and end of the active video and blanking fields. An activity detector checks for the transitions on the PDI bus. It samples the values on the PDI bus and once it has detected valid activity, sets a flag in the status register and can optionally trigger an interrupt. The PDI interface has a state machine which extracts the control information

from the video data. The machine checks the video data for the Preamble Field (0xFF, 0x00,0x00) and then depending on the status bits XY decides if it has received a valid control signal.

55.10.1.5 Normal and Narrow Mode

In normal mode, PDI support maximum input frequency of 32 MHz. In narrow mode, PDI supports maximum input frequency of 64 MHz.

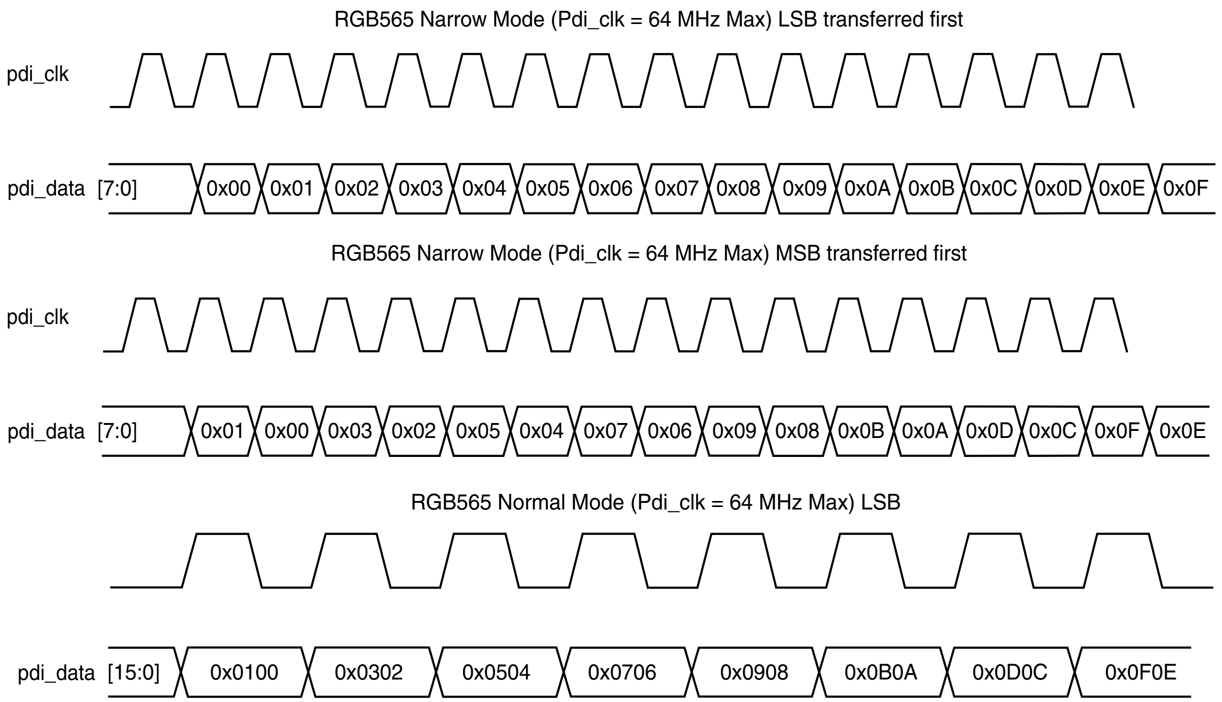


Figure 55-1958. Data Transfer in Normal and Narrow Mode

The byte transferred first (MSB or LSB) depends on the configuration register as shown in Figure 55-1958.This would not effect the sync preamble sequence in case internal sync mode.

On this device, the incoming RGB data is mapped onto the PDI pins as described in the following table.

Table 55-1954. Mapping of RGB data onto PDI pins

Mode	Mapping
Normal (full 18-bit PDI interface)	PDI[17:12] = DCU4_R[5:0] PDI[11:6] = DCU4_G[5:0] PDI[5:0] = DCU4_B[5:0]
Normal (RGB565 16-bit PDI interface)	PDI[15:11] = DCU4_R[4:0] PDI[[10:5] = DCU4_G[5:0] PDI[4:0] = DCU4_B[4:0]

Table continues on the next page...

Table 55-1954. Mapping of RGB data onto PDI pins (continued)

Mode	Mapping
Narrow (8-bit PDI interface)	RGB565: In first clock cycle, PDI[7:0] = { DCU4_R[4:0], DCU3_G[5:3] }. In second clock cycle, PDI[7:0] = { DCU3_G[2:0], DCU3_B[4:0] }. YCbCr: In first clock cycle, PDI[7:0] = { DCU3_Cb[7:0] }. In second clock cycle, PDI[7:0] = { DCU3_Y0[7:0] }. In third clock cycle, PDI[7:0] = { DCU3_Cr[7:0] }. In forth clock cycle, PDI[7:0] = { DCU3_Y1[7:0] }.

55.10.1.6 Modes of Operation Based on Sync Extraction

55.10.1.6.1 PDI input data (external sync mode)

In external sync mode the timing signals (HSYNC, VSYNC and, optionally, Date Enable) are provided to the PDI input timing pins by the external video source.

External sync mode can be used in both normal mode and 8-bit narrow mode, but cannot be used in conjunction with the YCbCr data format. In the instance that external sync and narrow mode is selected, the external signals are used, and any timing information (EAV/SAV) embedded in the data stream is ignored.As in Figure 11-95, PDI data enable (PDI_DE) should be low during VSYNC and HSYNC pulse, VSYNC front porch (FP_V) and back porch (BP_V), HSYNC front porch (FP_H) and back porch (BP_H). This is valid for Data Enable Mode when the PDI_DE_EN bit is set in the DCU_MODE register (i.e. mode with HSYNC, VSYNC, PDI_DE and PDI_PCLK as pin signals).

As in Figure 55-1960, PDI data enable (PDI_DE) should be low during VSYNC and HSYNC pulse, VSYNC front porch (FP_V) and back porch (BP_V), HSYNC front porch (FP_H) and back porch (BP_H). This is valid for Data Enable Mode when the PDI_DE_EN bit is set in the DCU_MODE register (i.e. mode with HSYNC, VSYNC, PDI_DE and PDI_PCLK as pin signals).

Pulse width, Front and back porch values should be picked from those programmed in DCU3 registers. In order to achieve lock, it must have same value as that of TFT screen. Front porch and back porch value can be zero. Pulse width and TFT screen size parameters cannot be zero. In case they are programmed as zero, it might lead to malfunctioning of the validation state machine.

As in Figure 55-1959 Figure 11-94 HSYNC must occur during the VSYNC and vertical blanking period. The time between 2 HSYNC should be the same during VSYNC and vertical blanking as during the active line period. As in Figure 11-94 the positive edge of HSYNC and VSYNC should be aligned. As in Figure 11-94 the positive edge of the HSYNC and start of the vertical front/back porch should be aligned.

The polarity of HSYNC and VSYNC is selectable.

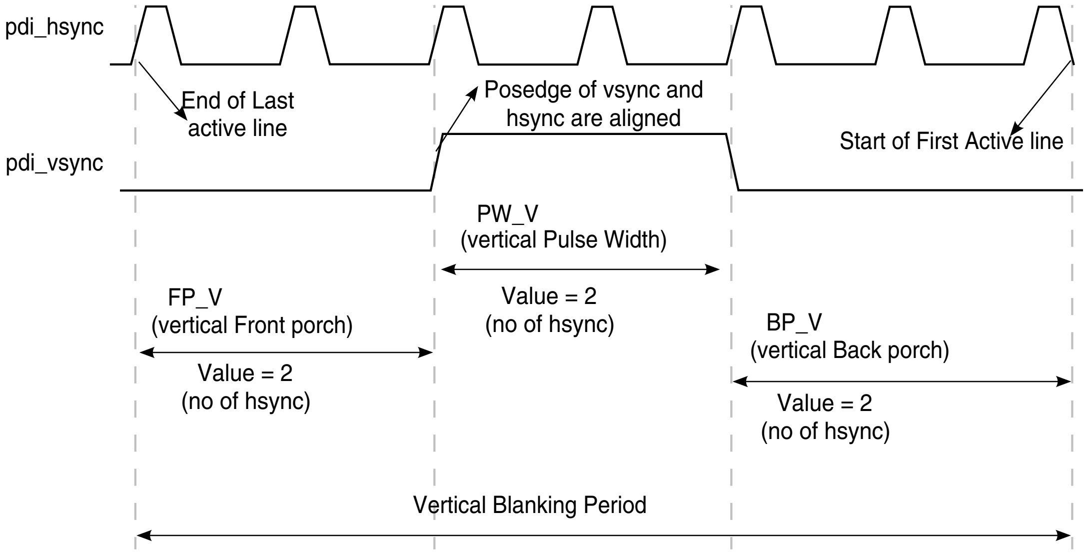


Figure 55-1959. Relation between Hsync and Vsync in external synchronization

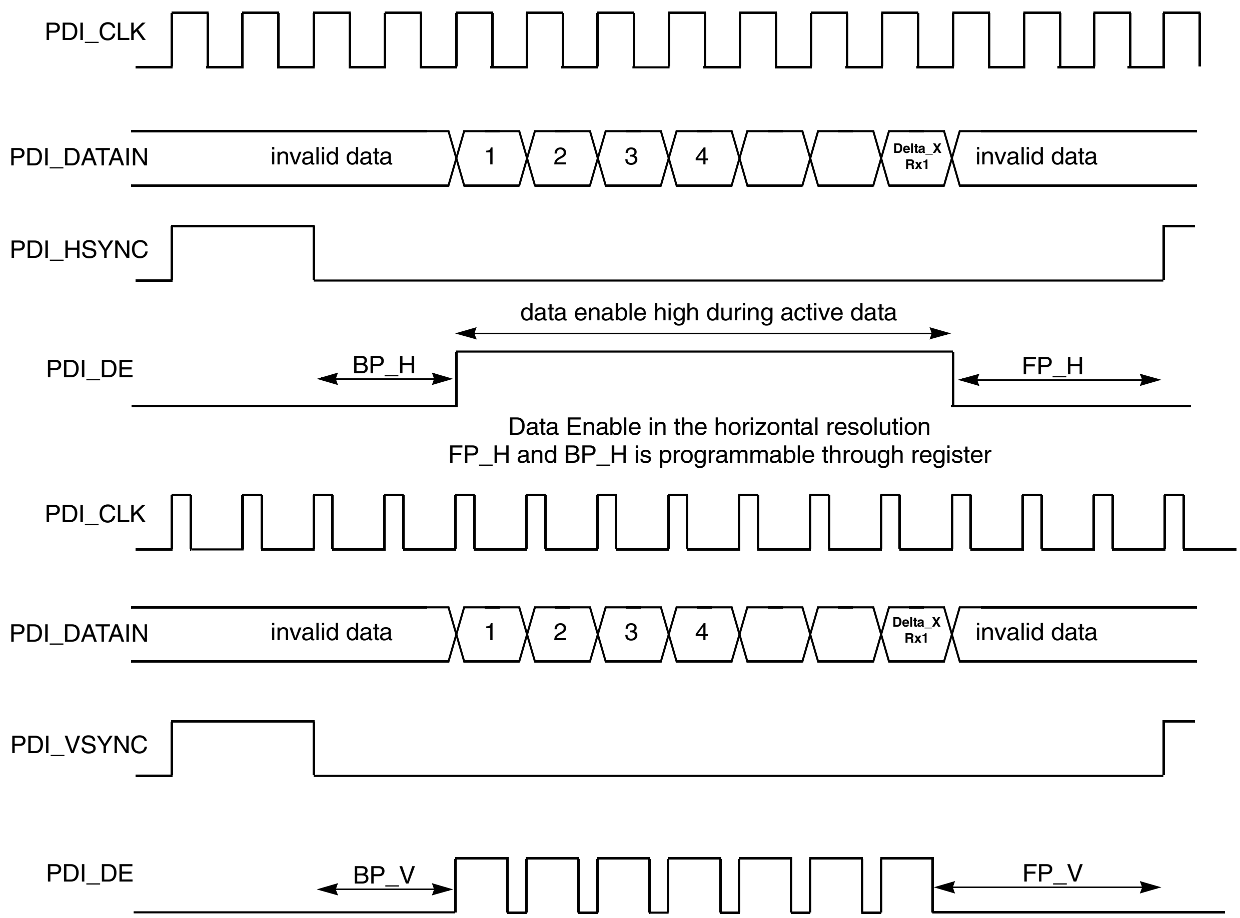


Figure 55-1960. Occurrence of Hsync and Vsync and DataEn for the entire frame

55.10.1.6.2 PDI Input data (Internal Sync Extraction mode)

In internal sync mode the timing parameters (horizontal and vertical blanking) are encoded into the data stream.

Internal sync mode can only be used in 8-bit narrow mode.

In Figure 55-1961, XY is used to decode the vertical and horizontal blanking period.

Table 55-1955. XYh Value

Bit	Value	Description
7	1'b1	Always 1'b1. This is checked while decoding sync preamble
6	F	Not considered in the state machine logic

Table continues on the next page...

Table 55-1955. XYh Value (continued)

Bit	Value	Description
5	V	1'b1 during vertical blanking 1'b0 elsewhere
4	H	1'b0 for start of active video 1'b1 for end of active video
3 2 1 0	P3 P2 P1 P0	Protection bits (used to detect ECC errors). It would not be used for bit correction.

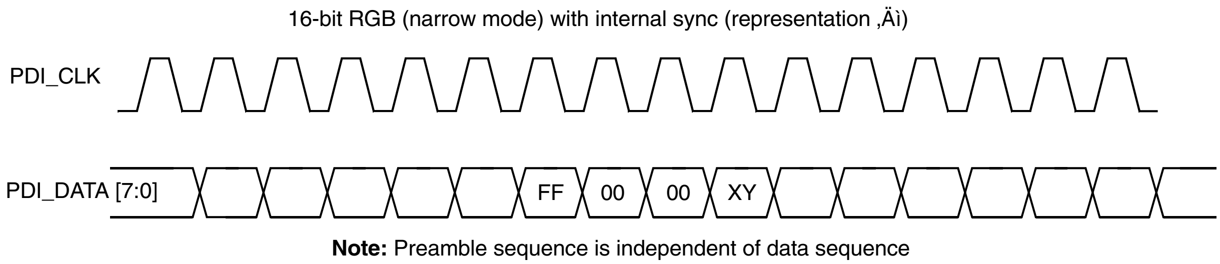


Figure 55-1961. Location of sync preamble in narrow mode

Sync Preamble would come continuously for 4 clock cycles as shown in Figure 55-1961. It does not depend upon which byte is coming first in data (MSB or LSB). Sync extraction is done using PDI_DATAIN[7:0] to identifies the horizontal and vertical blanking period using H and V field of the 'XYh' data as mentioned in Table 55-1955.

ITU 656 Sync preamble pattern (FFh 00h 00h) has to be masked out in the RGB and YCbCr data. The data stream must not include FFh 00h 00h as the valid pixel data to avoid malfunction by the validation state machine.

Horizontal blanking period must continue during the Vertical blanking period. The gap between 2 horizontal blanking periods should be the same during vertical blanking period as during line active. All Vertical and horizontal parameter values are validated against the DCU3 registers programmed by the user. Polarity of HSYNC and VSYNC are selectable. Horizontal blanking and vertical blanking must be aligned as shown in Figure

55-1962. During blanking period the input stream must be a 80h 10h 80h 10h sequence. This sequence must be present during both horizontal (line) blanking and vertical (frame) blanking.

As the PDI does not support interlaced video, the Framing Bit (F field in XYh) will be ignored during timing extraction. Narrow mode is compatible with RGB565 and YCbCr422 muxed modes. Each header contains an ECC value, which the PDI will check. The PDI is capable of detecting an error but not correcting it.

As with External sync mode, the value of front and back porch can be zero but the pulse width and TFT screen parameter cannot be zero.

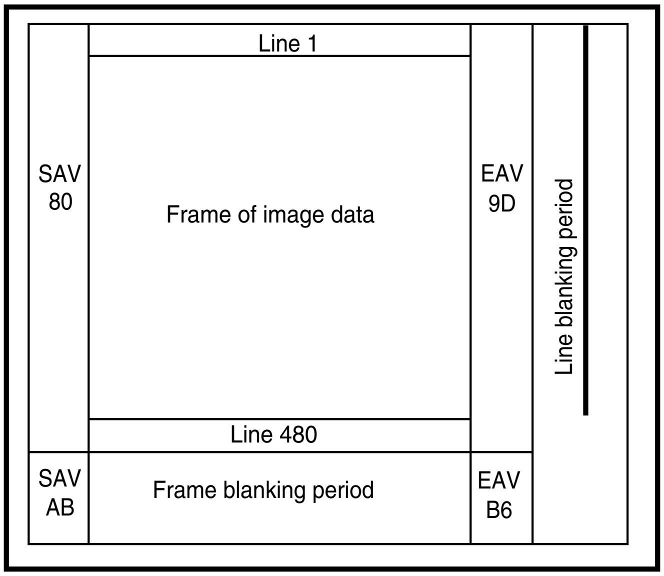


Figure 55-1962. Relationship between Hblank and Vblank in internal sync

55.10.1.6.3 PDI YCbCr mode and DCU4 YCbCr color format

The DCU4 can process incoming data from the PDI and from memory in YCbCr422 format. Both sources use the same RGB conversion and interpolation equations, however, the coefficients for the equations and enable for the interpolation are independently controlled.

In YCbCr mode, the PDI extracts the ITU656 sync (FF-00-00) and sends the video to the processing functions. The first processing function converts the 422 stream to a 444 stream, by providing interpolation on the chroma components of the stream depending on PDI_INTERPOL_EN bit. The second processing function converts the stream to RGB888/RGB565.

The DCU4 can also select YCbCr format in the BPP field of layer descriptor 4, which allows the same process to be applied to values stored in memory rather than brought in from the PDI. Interpolation is controlled for the layers using the LYR_INTPOL_EN register.

The RGB pixel value is computed using following equations:

$$\text{Red} = \frac{(Y-16).\text{yred}}{512} + \frac{(\text{Cr}-128)\text{Crred}}{512} + \frac{(\text{Cb}-128)\text{Cbred}}{512}$$

$$\text{Green} = \frac{(Y-16).\text{Ygreen}}{512} + \frac{(\text{Cr}-128)\text{Crgreen}}{512} + \frac{(\text{Cb}-128)\text{Cbgreen}}{512}$$

$$\text{Blue} = \frac{(Y-16).\text{yblue}}{512} + \frac{(\text{Cr}-128)\text{Crblue}}{512} + \frac{(\text{Cb}-128)\text{Cbblue}}{512}$$

NOTE

The first multiplication, ((y –16) * ycoeff), is unsigned, the two others are signed.

The register values after reset are as follows:

Yred = 10'h254 (596/512 = 1.16)

Crred = 11'h331 (817/512 = 1.6)

Cbred = 12'h000

Ygreen = 10'h254(596/512 = 1.16)

Crgreen = 11'h660(-416/512 = –0.812)

Cbgreen = 12'hf38(-200/512 = –0.39)

Yblue = 10'h254(596/512 = 1.16)

Crblue = 11'h000

Cbblue = 12'h409(1033/512 = 2.017)

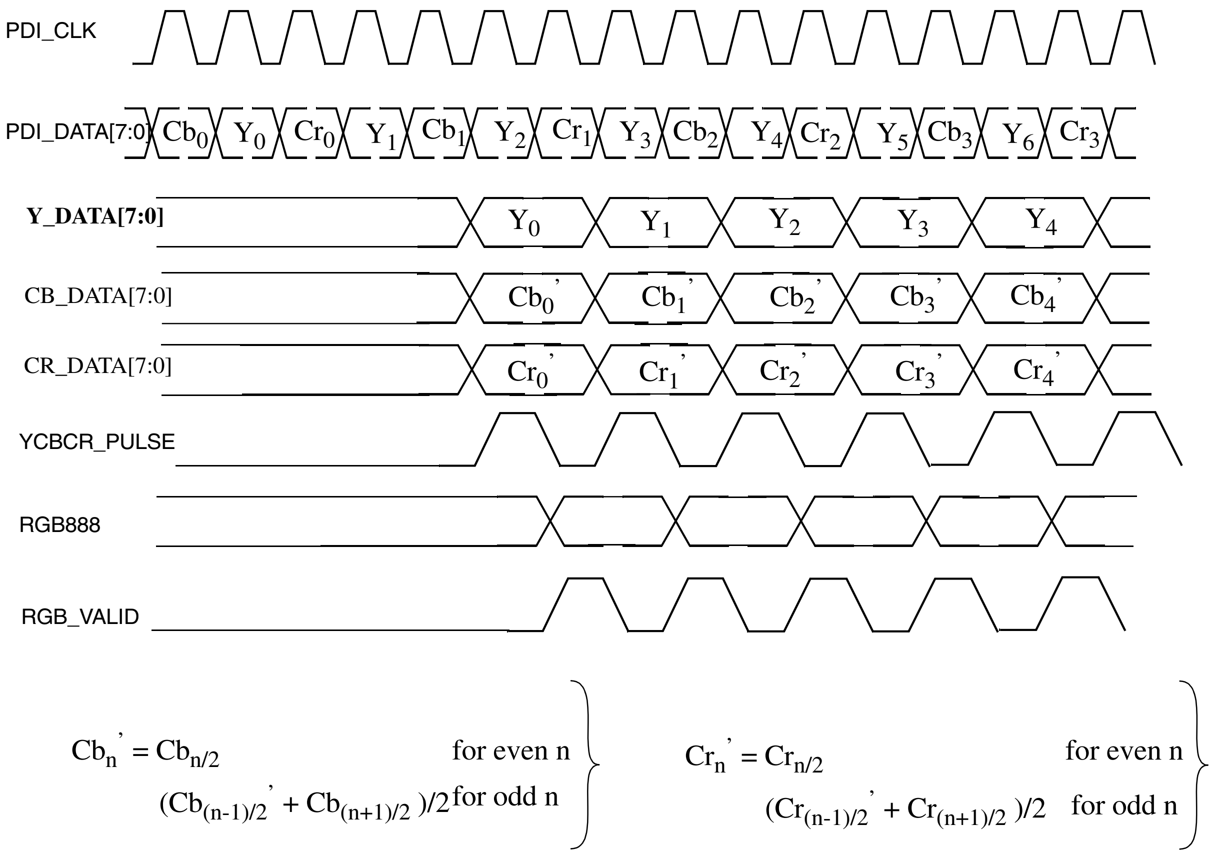


Figure 55-1963. YCbCr timing diagram

55.10.1.7 Mode of operation depending on PDI[17:0]

PDI supports following modes (other than the Slave Mode):

- 8-bit monochrome (8-bit input data, each pixel info is coming in 1 clock)
- 16-bit – RGB565 (16-bit input data, each pixel info is coming in 1 clock)
- 18-bit – RGB666 (18-bit input data, each pixel info is coming in 1 clock)
- 16-bit – RGB565 (8-bit input data, each pixel info is coming in 2 clocks)
- 16-bit – YCbCr422 (8-bit input data, info for 2 co-sited pixels coming in 4 clocks)

Data info extraction is given in Table 55-1956.

Table 55-1956. Data extraction in all possible modes

PDI Mode	Narrow Mode	Pins	Data	Notes
8-bit monochrome mode	1'b0	8 bit	PDI[7:0]	—
RGB565	1'b0	16 bit	PDI[15:0]	—
RGB666	1'b0	18 bit	PDI[17:0]	—
RGB565 muxed	1'b1	8 bit	PDI[7:0]	Data from two clocks are combined.
YCbCr422	1’b1	8 bit	PDI[7:0]	Data from four clocks are combined for 2 pixels.

The 8-bit monochrome image is equivalent to 8-bit grayscale images. For converting 8-bit monochrome data to RGB data, each of the R/G/B components will have a value equal to the 8-bit monochrome value.

RGB extraction starts when PDI is enabled (from the next falling edge of validated vertical blanking period)

55.10.1.8 PDI-Related Interrupts

PDI can be configured to trigger an interrupt when synchronization is achieved i.e. it receives the prespecified numbers of frames without error. PDI can also give an interrupt when synchronization is lost i.e. it receives any error in frame there after. This interrupt is raised when HSYNC/VSYNC is lost.

The PDI can also trigger an interrupt if there is either a one bit or a multiple bit error in the ECC value during the extraction of the preamble in internal synchronization mode.

Blanking sequence error interrupt can be triggered in case 80h 10h is not found in vertical and line blanking period during internal synchronization.

Activity detection interrupt for CLK detection.

HSYNC, VSYNC and DE detection interrupts can be set to generate upon transitions on the PDI_HSYNC, PDI_VSYNC and PDI_DE pins.

Activity lost interrupt for PDI_CLK — for the PDI_CLK lost interrupt, is triggered when the PDI_CLK is less than DCU3 module clock frequency divided by 32. (i.e. if DCU3 module clock freq. max = 64 MHz, then the PDI_CLK_LOST flag will be set if pdi clk freq. min < 2 MHz).

All interrupts are RW1C (write one to clear). All interrupts are maskable.

PDI must reset to show the latest status of the clock activity detect interrupt.

55.10.2 Switch between DCU mode and PDI mode (top-level description)

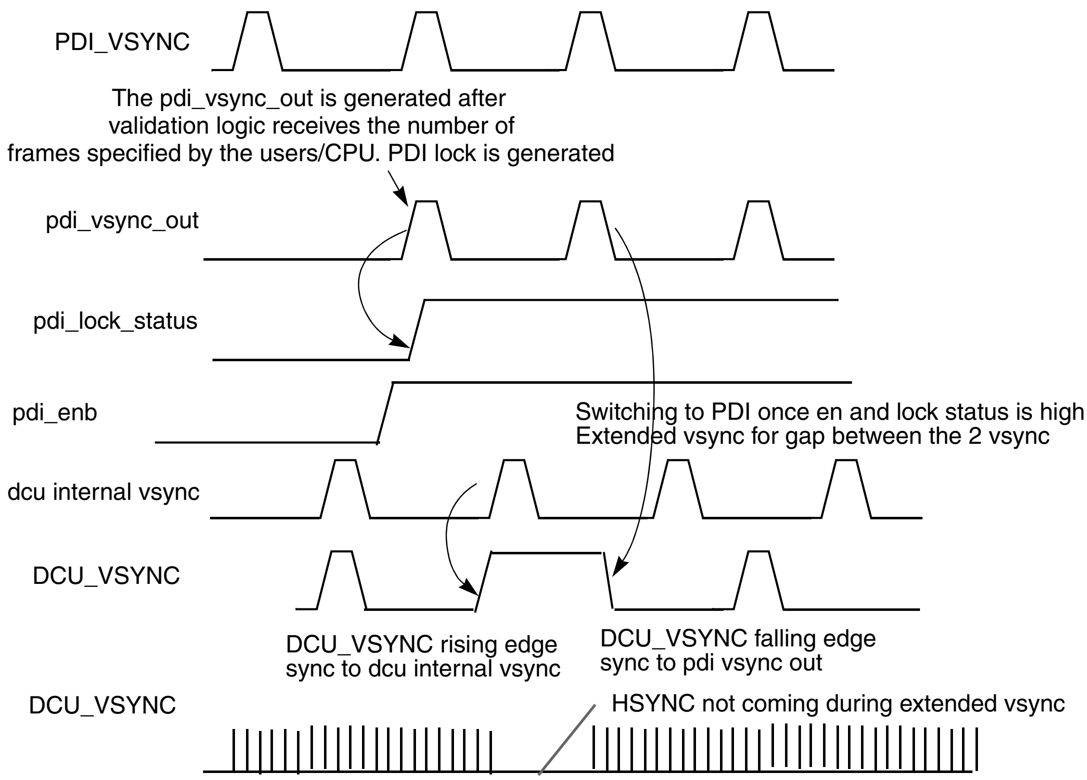


Figure 55-1964. Switch to PDI on receiving the interrupt

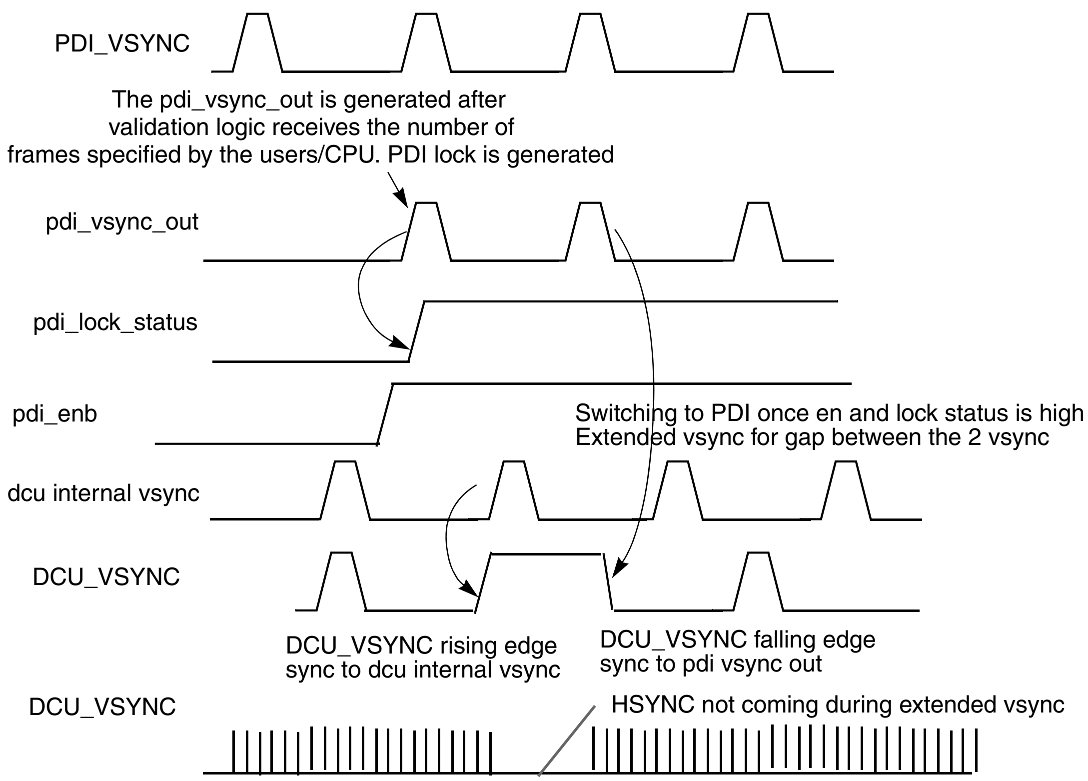


Figure 55-1965. Switch to Normal Mode from PDI mode

55.10.2.1 Changes in the configuration

Any changes in the RGB format or the synchronization mode configuration requires the PDI system to be in disable mode i.e. PDI_EN should be 0 during this time. The resolution of the screen/layer (PDI or TFT) cannot change on the fly.

55.10.2.1.1 PDI slave mode

The VSYNC generated by the DCU4 block is synchronized to the PDI input VSYNC; the PDI VSYNC resets the internal timing generation unit. HSYNC and VSYNC are generated internally corresponding to the external TFT screen display parameters programmed in the DCU4 registers.

55.10.2.1.2 PDI sync detection/validation

PDI declares a lock when it has correctly received the continuous number of frames programmed by the application.

PDI declares the sync lost:

- When it receives data enable during the vertical and horizontal blanking period (in case of data enable mode).
- When the incoming HSYNC and VSYNC timing does not match the programmed parameter values in the DCU3. This also means that if any of control signal info lost, then it also declares sync lost.

After PDI_PCLK is lost, the PDI requires 64 DCU3 module clock cycles to detect PDI_PCLK again.

PDI does not declare lost sync in case of blanking and ECC errors.

Writes to the input FIFO are stopped as soon as PDI sync is lost aborting transfer of the current frame data. Sync detection works continuously, independent of PDI enable. PDI fires an interrupt on:

- Sync lock is achieved
- Sync is lost
- Activity is detected on hsync
- Activity is detected on vsync
- PCLK activity detection (it is not generated from the state machine)
- DE Activity detection (it is not generated from the state machine)
- PCLK activity lost (it is not generated from the state machine)

On receiving a wrong sync pulse, the DCU4 stops the HSYNC and VSYNC activity detection and gives an interrupt when it finds sync pulse again. The state machine works for zero values of front and back porches but not for pulse width and screen parameters.

55.10.2.1.3 Other assumptions

The reset to the PDI clock is synchronized to the peripheral clock. Reset synchronization is done with respect to the PDI clock internally.

The PDI clock should be available at least 10 clock after the last valid data. This corresponds to the delay of the PDI block.

55.11 DCU4 Initialization

The following steps describe a typical approach to initializing the DCU4 for use in an application.

1. After reset, configure the DCU4 peripheral to be active and configure the DCU4 clock source.
2. If using a panel with an integrated TCON module, disable the TCON signals by setting the TCON_BYPASS bit in the TCON CTRL1 register. Due to the configuration of the TCON module, the DCU4 pixel clock signal will be output as soon as it is selected by the related IOMUX register. This is independent of the DCU4 operating mode.
3. Configure the output ports in the IOMUX registers as required.
4. Configure the timing registers to match the TFT LCD panel in use (see TFT LCD panel configuration).
5. Set the background color as required.
6. Load the initial tile or palette colors into the CLUT/Tile memory.
7. Initiate a manual refresh of the frame by setting the READREG bit in the UPDATE_MODE register.
8. If Automatic transfer is required, wait until the READREG bit is cleared before setting the MODE bit in the UPDATE_MODE register

55.12 Glossary

Table 55-1957. Glossary

ARGB (also BGRA)	A data format where the pixel values are stored using four components: Alpha, Red, Green and Blue. DCU4 supports different variations of this format where different numbers of bits can be used to represent each of the components
Component	Part of a pixel that contains a single color (red, green or blue)
CLUT	Color Look-up table. The table that contains the palette used by an indexed-color graphic
Direct color	The full 24-bit value actually written to a pixel to create a color
Frame	The collection of all pixels on a panel
Gamut	The set of colors that a panel can display. In most cases a panel cannot display the full gamut of colors visible to the human eye.
Indexed color	An index into a table containing direct-colors. Usually smaller in size than the direct color; the DCU4 provides 1, 2, 4, and 8 bits per pixel options

Table continues on the next page...

Table 55-1957. Glossary (continued)

Palette	The list of colors used by a graphic when an indexed colors format is used. The palette is stored in a color look up table and can be from one color up to the maximum of the size of the CLUT.
Panel	A TFT LCD containing an array of colored pixels.
Pixel	The basic graphical element on a TFT LCD panel. Can display a range of colors depending on the value of the red, green and blue values written to it. Normally arranged in a rectangular array.
RGB	A data format where the pixel values are stored using three components: Red, Green and Blue. DCU4 supports different variations of this format where different numbers of bits can be used to represent each of the components
Vertical blanking period	A time during the TFT LCD panel refresh cycle when no data is being written to the panel

Chapter 56
LCD Driver LCD64F6B

56.1 LCD Driver (LCD64F6B)

56.2 Information Specific to This Device

This section presents device-specific parameterization and customization information not specifically referenced in the remainder of this chapter.

56.2.1 Number of Front and Back Planes

Table 56-1. Number of Front and Back Planes

Parameter	Value
Number of front planes[1]	36 or 38 or 40
Number of back planes[2]	4 or 6 or 8

1. Software-configurable
2. Software-configurable

56.2.2 LCD Clock Selection

The following table shows the clocks selected by the LCDCR[LCDOCS] bit.

Table 56-2. LCD Clock Selection Based on LCDCR[LCDOCS]

Value of LCDCR[LCDOCS]	Clock selected
1	32 kHz OSC
0	128 kHz OSC

56.2.3 Settings during STANDBY mode

To keep the LCD driver shut down in STANDBY mode, the following settings are needed:

- LCDCR[LCDRST] = 0
- LCDCR[LCDRCS] = 0

To keep the LCD driver on (functioning) in STANDBY mode, the following settings are needed:

- LCDCR[LCDRST] = 1
- LCDCR[LCDRCS] = 1
- LCDCR[LCDOCS] = 0 or 1
  - If this field is 0, the LCD driver will operate from SIRC.
  - If this field is 1, the LCD driver will operate from SXOSC.

56.3 Introduction

This section introduces the LCD driver module.

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

56.3.1 Overview

The LCD driver module has up to 64 frontplane drivers (*n*) and up to 8 backplane drivers (*m*) so that a maximum of 512 LCD segments are controllable. The actual implementation (*n,m*) depends on the device specification. Each segment is controlled by a corresponding bit in the LCD RAM. *m* multiplex modes (1/1, 1/2,…1/*m* duty), and three bias (1/1, 1/2, 1/3) methods are available. The V_0 voltage is the lowest level of the output waveform and V_3 becomes the highest level. All frontplane and backplane pins can be multiplexed with other Port functions.

The LCD driver system consists of five major sub-modules:

- Timing and Control – consists of registers and control logic for frame clock generation, bias voltage level select, frame duty select, contrast adjustment, backplane select and frontplane select/enable, remapping of backplane drivers to produce the required frame frequency and voltage waveforms.
- LCD RAM – contains the data to be displayed on the LCD. Data can be read from or written to the display RAM at any time.
- Frontplane Drivers – consists of *n* frontplane drivers.
- Backplane Drivers – consists of *m* backplane drivers.
- Voltage Generator – Based on reference voltage VDDE. It generates the voltage levels for the timing and control logic to produce the frontplane and backplane waveforms.

56.3.2 Block Diagram

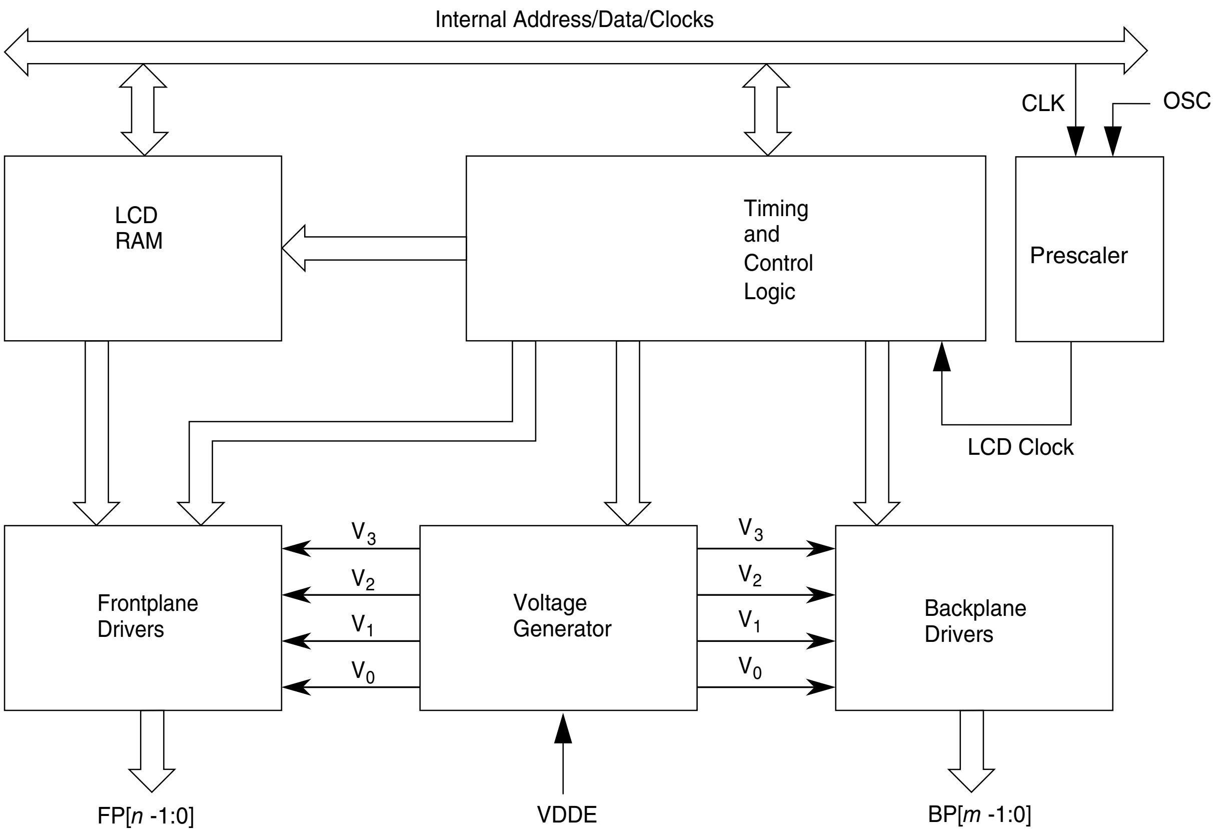


Figure 56-1. Block Diagram

56.3.3 Features

The LCD64F6B includes these distinctive features:

- Up to 64 frontplane drivers
  - Each frontplane has an individual enable bit
- Up to 8 backplane drivers
- Remapping of backplane drivers
- Programmable frame clock generator
- Programmable bias voltage level selector
- Programmable output current
- Selectable output current boost during transitions.
- Selectable LCD frame frequency interrupt event
- On-chip generation of four different output voltage levels
- Contrast adjustment by using Contrast adjustment phases
- Selectable continuous drive of LCD while in power down mode

56.3.4 Modes of Operation

The LCD64F6B module supports up to seven operation modes with different numbers of backplanes and different biasing levels. During power saving mode the LCD operation can be suspended under software control. Depending on the state of internal bits, the LCD can operate normally with source clock applied, or the LCD clock generation can be turned off and the LCD64F6B module enters a power conservation state.

This is a high level description only, detailed descriptions of operating modes are contained in later sections.

56.4 External Signal Description

Table 56-3. Signal Properties

Name	Port	Function	Reset	Pull Up
m Backplane Waveforms	BP[*m*-1:0]	Backplane waveform signals that connect directly to the pads	high impedance	
n Frontplane Waveforms	FP[*n*-1:0]	Frontplane waveform signals that connect directly to the pads	high impedance	
VDD33 Voltage	VDD33	LCD supply and reference voltage		
VSSE Voltage	VSSE	LCD ground voltage		

56.4.1 Detailed Signal Descriptions

Table 56-4. Interface A—Detailed Signal Descriptions

Signal	Description
BP[*m*-1:0]	This output signal vector represents the analog backplane waveforms of the LCD64F6B module and is connected directly to the corresponding pads.
FP[*n*-1:0]	This output signal vector represents the analog frontplane waveforms of the LCD64F6B module and is connected directly to the corresponding pads.
VDDE	Positive supply and reference voltage for the LCD waveform generation.
VSSE	Ground supply voltage for the LCD waveform generation.

56.5 Memory Map and Registers

This section describes the Memory map table and the register descriptions in address order.

LCD memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_E000	LCD Control Register (LCD_LCDCR)	32	R/W	0000_0000h	56.5.1/3230

Table continues on the next page...

LCD memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_E004	LCD Prescaler Control Register (LCD_LCDPCR)	32	R/W	0000_0000h	56.5.2/3232
400B_E008	LCD Contrast Control Register (LCD_LCDCCR)	32	R/W	0000_0000h	56.5.3/3233
400B_E010	LCD Frontplane Enable Register 0 (LCD_ENFPR0)	32	R/W	0000_0000h	56.5.4/3233
400B_E014	LCD Frontplane Enable Register 1 (LCD_ENFPR1)	32	R/W	0000_0000h	56.5.5/3234
400B_E020	LCDRAM (LCD_Location 0)	32	R/W	0000_0000h	56.5.6/3234
400B_E024	LCDRAM (LCD_Location 1)	32	R/W	0000_0000h	56.5.7/3235
400B_E028	LCDRAM (LCD_Location 2)	32	R/W	0000_0000h	56.5.8/3236
400B_E02C	LCDRAM (LCD_Location 3)	32	R/W	0000_0000h	56.5.9/3236
400B_E030	LCDRAM (LCD_Location 4)	32	R/W	0000_0000h	56.5.10/ 3237
400B_E034	LCDRAM (LCD_Location 5)	32	R/W	0000_0000h	56.5.11/ 3238
400B_E038	LCDRAM (LCD_Location 6)	32	R/W	0000_0000h	56.5.12/ 3238
400B_E03C	LCDRAM (LCD_Location 7)	32	R/W	0000_0000h	56.5.13/ 3239
400B_E040	LCDRAM (LCD_Location 8)	32	R/W	0000_0000h	56.5.14/ 3240
400B_E044	LCDRAM (LCD_Location 9)	32	R/W	0000_0000h	56.5.15/ 3240

56.5.1 LCD Control Register (LCD_LCDCR)

Address: 400B_E000h base + 0h offset = 400B_E000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	LCDEN	LCDRST	LCDRCS	DUTY			BIAS	VLCDS	PWR		BSTEN	BSTSEL	BSTAO	LCDOCS	LCDINT	EOF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NOF								0	0	LCDBPA		0	LCDBPS		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_LCDCR field descriptions

Field	Description
31 LCDEN	LCD Driver System Enable. 0 All frontplane and backplane pins are disabled. In addition, the LCD Driver is disabled and all LCD waveform generation clocks are stopped. 1 LCD Driver System is enabled. All FP[n-1:0] pins with ENFP[n-1] set, will output an LCD driver waveform. The BP[m-1:0] pins will output an LCD Driver waveform based on the settings of DUTY.
30 LCDRST	Continue to drive LCD display while stop/standby requested.Can be written only when LCDEN is cleared. See Section “Operation in Power Saving Modes” for details. 0 Stop LCD64F6B display driver system while stop/standby requested. 1 LCD display driver system operates normally while stop/standby requested.
29 LCDRCS	LCD Reference Clock Select.Can be written only when LCDEN is cleared.It controls the selection between oscillator clock and system clock. 0 System clock is selected. 1 scillator clock selected based on LCDOCS.
28–26 DUTY	LCD Duty Select. The DUTY bits select the duty (multiplex mode) of the LCD Driver system as shown in Table 56-23 . The multiplex mode should be changed only when LCD Driver is disabled, LCDEN is not set.
25 BIAS	BIAS Voltage Select.This bit selects the bias voltage levels during various LCD operating modes, as shown in Table 56-23
24 VLCDS	Reserved Can be written only when LCDEN is cleared. See Operation in Power Saving Modes 0 No effect. 1 No effect.
23–22 PWR	LCD Power mode The PWR bits select the output current and have a direct impact on the power consumption and drive capability of the LCD64F6B module. Table 56-24 lists the possible selections.
21 BSTEN	LCD output current boost enable Since the LCD appears like a capacitance to the driver it is sometimes useful to boost the output current during transitions to increase the slew rate of the driver. 0 No output current boost available. 1 Output current boost during transitions according to the BSTSEL bit.
20 BSTSEL	LCD output current boost select The BSTSEL bit sets the multiplier for the output current boost. If the BSTEN bit is set, the output current is boosted during transitions in the following way: 0 Current boosting by a factor of 8. 1 Current boosting by a factor of 16.
19 BSTAO	LCD Boost always on.If set, the selected by BST and enabled by BSTEN boost current is always on and not only during the time frame generated by the state machine 0 The Boost always on feature is disabled. 1 The Boost always on feature is enabled.
18 LCDOCS	LCD OSC Clock Select. Can be written only when LCDEN is cleared..It selects between oscillator clocks. 0 128Khz Slow Internal RC Oscillator Clock is selected. 1 32Khz External Oscillator clock is selected.
17 LCDINT	LCD interrupt enable. 0 Interrupt request is disabled. 1 Interrupt will be requested whenever EOF is set.

Table continues on the next page...

LCD_LCDCR field descriptions (continued)

Field	Description
16 EOF	End of frame interrupt flag. End of frame interrupt flag bit is set every time when NOF frames have been executed while LCDEN is set. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LCDINT = 1), EOF causes an interrupt request. 0 Defined via NOF bits frames have not been executed yet. 1 Defined via NOF bits frames have been executed.
15–8 NOF	Number of frames.The number of frames bits determine how many frames are counted until an interrupt will be requested. The possible number of frames are: 0x00 An interrupt is requested at the end of every frame 0x01 An interrupt is requested at the end of every second frame. .... and so on up to 0xFF An interrupt is requested at the end of every 256th frame
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 LCDBPA	Backplane Adding. If set, the backplanes, which are not available in standard configuration, will be mapped on frontplanes as shown in Table 469. Hence, up to six backplanes can be used to drive LCD displays.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 LCDBPS	Backplane Shifting. Using these bits, backplanes will be swapped with frontplanes as shown in Table 469.

56.5.2 LCD Prescaler Control Register (LCD_LCDPCR)

Address: 400B_E000h base + 4h offset = 400B_E004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				LCLK				0																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_LCDPCR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–24 LCLK	LCD Clock Prescaler. The LCD Clock Prescaler bits determine the clock divider value to produce the LCD Clock Frequency. For detailed description of the correlation between LCD Clock Prescaler bits and the divider value please refer to Table 56-21. LCD Clock Prescaler bits should be changed only when LCD Driver is disabled, LCDEN is not set.
23–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

56.5.3 LCD Contrast Control Register (LCD_LCDCCR)

Address: 400B_E000h base + 8h offset = 400B_E008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	CCEN	Reserved				LCC										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_LCDCCR field descriptions

Field	Description
31 CCEN	LCD Contrast Control Enable. 0 The Contrast Control is disabled. 1 The Contrast Control is enabled. The length of the contrast phase depends on the bits LCC.
30–27 Reserved	This field is reserved. Reserved.
26–16 LCC	LCD Contrast Control. The Contrast Control bits determine the width of the contrast phase. 0x7FF=Contrast Phase lasts the whole duty cycle. 0x000=Contrast Phase has no duration what results in highest contrast.
15–0 Reserved	This field is reserved. Reserved.

56.5.4 LCD Frontplane Enable Register 0 (LCD_ENFPR0)

Address: 400B_E000h base + 10h offset = 400B_E010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	ENFP[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_ENFPR0 field descriptions

Field	Description
31–0 ENFP[31:0]	Frontplane Output Enable. The ENFP[31:0] bits enable the frontplane driver outputs. If LCDEN = 0, these bits have no effect on the state of the I/O pins. It is recommended to set ENFP[31:0] bits before LCDEN is set. 1 Frontplane driver output enabled on FP[31:0]. 0 Frontplane driver output disabled on FP[31:0].

56.5.5 LCD Frontplane Enable Register 1 (LCD_ENFPR1)

Address: 400B_E000h base + 14h offset = 400B_E014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENFP[63:32]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_ENFPR1 field descriptions

Field	Description
31–0 ENFP[63:32]	Frontplane Output Enable. The ENFP[63:32] bits enable the frontplane driver outputs. If LCDEN = 0, these bits have no effect on the state of the I/O pins. It is recommended to set ENFP[63:32] bits before LCDEN is set. 1 Frontplane driver output enabled on FP[63:32]. 0 Frontplane driver output disabled on FP[63:32]. **NOTE:** The implemented ENFP[n-1] bits depend on the number of implemented frontplanes.

56.5.6 LCDRAM (LCD_Location 0)

Address: 400B_E000h base + 20h offset = 400B_E020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP0_BP[7:0]								FP1_BP[7:0]								FP2_BP[7:0]								FP3_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 0 field descriptions

Field	Description
31–24 FP0_BP[7:0]	LCD segment ON. The FP0BP[7:0] bit displays (turns on) the LCD segment connected between FP0and BP[7:0].

Table continues on the next page...

LCD_Location 0 field descriptions (continued)

Field	Description
	1 LCD segment ON 0 LCD segment OFF
23–16 FP1_BP[7:0]	LCD segment ON. The FP1BP[7:0] bit displays (turns on) the LCD segment connected between FP1 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP2_BP[7:0]	LCD segment ON. The FP2BP[7:0] bit displays (turns on) the LCD segment connected between FP2 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
7–0 FP3_BP[7:0]	LCD segment ON.The FP3BP[7:0] bit displays (turns on) the LCD segment connected between FP3 and BP[7:0] 1 LCD segment ON 0 LCD segment OFF.

56.5.7 LCDRAM (LCD_Location 1)

Address: 400B_E000h base + 24h offset = 400B_E024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP4_BP[7:0]								FP5_BP[7:0]								FP6_BP[7:0]								FP7_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 1 field descriptions

Field	Description
31–24 FP4_BP[7:0]	LCD segment ON. The FP4BP[7:0] bit displays (turns on) the LCD segment connected between FP4and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
23–16 FP5_BP[7:0]	LCD segment ON. The FP5BP[7:0] bit displays (turns on) the LCD segment connected between FP5 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP6_BP[7:0]	LCD segment ON. The FP6 BP[7:0] bit displays (turns on) the LCD segment connected between FP6 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
7–0 FP7_BP[7:0]	LCD segment ON.The FP7BP[7:0] bit displays (turns on) the LCD segment connected between FP7 and BP[7:0].

Table continues on the next page...

LCD_Location 1 field descriptions (continued)

Field	Description
	1 LCD segment ON 0 LCD segment OFF.

56.5.8 LCDRAM (LCD_Location 2)

Address: 400B_E000h base + 28h offset = 400B_E028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP8_BP[7:0]								FP9_BP[7:0]								FP10_BP[7:0]								FP11_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 2 field descriptions

Field	Description
31–24 FP8_BP[7:0]	LCD segment ON. The FP8BP[7:0] bit displays (turns on) the LCD segment connected between FP8 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
23–16 FP9_BP[7:0]	LCD segment ON. The FP9BP[7:0] bit displays (turns on) the LCD segment connected between FP9 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP10_BP[7:0]	LCD segment ON. The FP10BP[7:0] bit displays (turns on) the LCD segment connected between FP10 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
7–0 FP11_BP[7:0]	LCD segment ON.The FP11BP[7:0] bit displays (turns on) the LCD segment connected between FP11 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF.

56.5.9 LCDRAM (LCD_Location 3)

Address: 400B_E000h base + 2Ch offset = 400B_E02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP12_BP[7:0]								FP13_BP[7:0]								FP14_BP[7:0]								FP15_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 3 field descriptions

Field	Description
31–24 FP12_BP[7:0]	LCD segment ON.The FP12BP[7:0] bit displays (turns on) the LCD segment connected between FP12 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
23–16 FP13_BP[7:0]	LCD segment ON. The FP13BP[7:0] bit displays (turns on) the LCD segment connected between FP13 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP14_BP[7:0]	LCD segment ON. The FP14BP[7:0] bit displays (turns on) the LCD segment connected between FP14 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
7–0 FP15_BP[7:0]	LCD segment ON.The FP15BP[7:0] bit displays (turns on) the LCD segment connected between FP15 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF.

56.5.10 LCDRAM (LCD_Location 4)

Address: 400B_E000h base + 30h offset = 400B_E030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP16_BP[7:0]								FP17_BP[7:0]								FP18_BP[7:0]								FP19_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 4 field descriptions

Field	Description
31–24 FP16_BP[7:0]	LCD segment ON.The FP16BP[7:0] bit displays (turns on) the LCD segment connected between FP16 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
23–16 FP17_BP[7:0]	LCD segment ON. The FP17BP[7:0] bit displays (turns on) the LCD segment connected between FP17 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP18_BP[7:0]	LCD segment ON. The FP18BP[7:0] bit displays (turns on) the LCD segment connected between FP18 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF

Table continues on the next page...

LCD_Location 4 field descriptions (continued)

Field	Description
7–0 FP19_BP[7:0]	LCD segment ON.The FP19BP[7:0] bit displays (turns on) the LCD segment connected between FP19 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF.

56.5.11 LCDRAM (LCD_Location 5)

Address: 400B_E000h base + 34h offset = 400B_E034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP20_BP[7:0]								FP21_BP[7:0]								FP22_BP[7:0]								FP23_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 5 field descriptions

Field	Description
31–24 FP20_BP[7:0]	LCD segment ON. The FP20BP[7:0] bit displays (turns on) the LCD segment connected between FP20and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
23–16 FP21_BP[7:0]	LCD segment ON. The FP21BP[7:0] bit displays (turns on) the LCD segment connected between FP21 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP22_BP[7:0]	LCD segment ON.The FP22BP[7:0] bit displays (turns on) the LCD segment connected between FP22 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
7–0 FP23_BP[7:0]	LCD segment ON.The FP23BP[7:0] bit displays (turns on) the LCD segment connected between FP23 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF.

56.5.12 LCDRAM (LCD_Location 6)

Address: 400B_E000h base + 38h offset = 400B_E038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP24_BP[7:0]								FP25_BP[7:0]								FP26_BP[7:0]								FP27_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 6 field descriptions

Field	Description
31–24 FP24_BP[7:0]	LCD segment ON. The FP24BP[7:0] bit displays (turns on) the LCD segment connected between FP24and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
23–16 FP25_BP[7:0]	LCD segment ON. The FP25 BP[7:0] bit displays (turns on) the LCD segment connected between FP25 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP26_BP[7:0]	LCD segment ON. The FP26 BP[7:0] bit displays (turns on) the LCD segment connected between FP26 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
7–0 FP27_BP[7:0]	LCD segment ON.The FP27 BP[7:0] bit displays (turns on) the LCD segment connected between FP27 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF.

56.5.13 LCDRAM (LCD_Location 7)

Address: 400B_E000h base + 3Ch offset = 400B_E03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP28_BP[7:0]								FP29_BP[7:0]								FP30_BP[7:0]								FP31_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 7 field descriptions

Field	Description
31–24 FP28_BP[7:0]	LCD segment ON. The FP28 BP[7:0] bit displays (turns on) the LCD segment connected between FP28and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
23–16 FP29_BP[7:0]	LCD segment ON. The FP29 BP[7:0] bit displays (turns on) the LCD segment connected between FP29 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP30_BP[7:0]	LCD segment ON. The FP30 BP[7:0] bit displays (turns on) the LCD segment connected between FP30 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF

Table continues on the next page...

LCD_Location 7 field descriptions (continued)

Field	Description
7–0 FP31_BP[7:0]	LCD segment ON.The FP31BP[7:0] bit displays (turns on) the LCD segment connected between FP31 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF.

56.5.14 LCDRAM (LCD_Location 8)

Address: 400B_E000h base + 40h offset = 400B_E040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP32_BP[7:0]								FP33_BP[7:0]								FP34_BP[7:0]								FP35_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 8 field descriptions

Field	Description
31–24 FP32_BP[7:0]	LCD segment ON. The FP32BP[7:0] bit displays (turns on) the LCD segment connected between FP32and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
23–16 FP33_BP[7:0]	LCD segment ON. The FP33BP[7:0] bit displays (turns on) the LCD segment connected between FP33 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP34_BP[7:0]	LCD segment ON. The FP34BP[7:0] bit displays (turns on) the LCD segment connected between FP34 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
7–0 FP35_BP[7:0]	LCD segment ON.The FP35BP[7:0] bit displays (turns on) the LCD segment connected between FP35 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF.

56.5.15 LCDRAM (LCD_Location 9)

Address: 400B_E000h base + 44h offset = 400B_E044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FP36_BP[7:0]								FP37_BP[7:0]								FP38_BP[7:0]								FP39_BP[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCD_Location 9 field descriptions

Field	Description
31–24 FP36_BP[7:0]	LCD segment ON. The FP36BP[7:0] bit displays (turns on) the LCD segment connected between FP36 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
23–16 FP37_BP[7:0]	LCD segment ON. The FP37 BP[7:0] bit displays (turns on) the LCD segment connected between FP37 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
15–8 FP38_BP[7:0]	LCD segment ON. The FP38BP[7:0] bit displays (turns on) the LCD segment connected between FP38 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF
7–0 FP39_BP[7:0]	LCD segment ON.The FP39 BP[7:0] bit displays (turns on) the LCD segment connected between FP39 and BP[7:0]. 1 LCD segment ON 0 LCD segment OFF.

56.6 Functional Description

This section describes frontplane, backplane, LCD System During Reset, LCD clock and frame frequency, contrast adjustment, LCD driver system enable and frontplane enable sequencing.

56.6.1 Frontplane, Backplane, and LCD System During Reset

During a reset the following conditions exist:

- All frontplane enable bits, ENFP[*n*-1:0] are cleared and the ON/OFF control for the display, the LCDEN bit is cleared, thereby forcing all frontplane and backplane driver outputs to the high impedance state. The pin state during reset is defined by the port control module.
- The LCD64F6B system is configured in the default mode, 1/1 duty and 1/1 bias, that means only BP0 is used, system clock as reference.

56.6.2 LCD Clock and Frame Frequency

The frequency of the clock and the clock divider determine the LCD Clock Frequency. The input clock for the prescaler can be selected by LCDRCS bit. The divider is set by the LCD Clock Prescaler bits in the LCD Prescaler Control Register as shown in the following table.

Table 56-21. Clock Divider

LCLK	Divider
0000	480
0001	2 * 480
0010	2^2 * 480
0011	2^3 * 480
0100	2^4 * 480
0101	2^5 * 480
0110	2^6 * 480
0111	2^7 * 480
1000	2^8 * 480
1001	2^9 * 480
1010	2^{10} * 480
1011	2^{11} * 480
1100	2^{12} * 480
1101	2^{13} * 480
1110	2^{14} * 480
1111	2^{15} * 480

The following formula may be used to calculate the LCD frame frequency:

$$\mathrm{LCDFrameFrequency}(\mathrm{Hz}) = \left[\frac{\mathrm{OSCCLK}(\mathrm{Hz})}{\mathrm{Divider}}\right]$$

Example: Clock = 16 MHz, Prescaler = 1010;

$$\left[\frac{16 \bullet 10^6}{2^{10} \bullet 480}\right] \approx 33\mathrm{Hz}$$

Note

A "Frame" is the full refresh cycle of the display. See LCD Waveform Examples for waveform illustrations.

56.6.3 Contrast Adjustment

The LCD Driver module offers two different ways to adjust contrast:

56.6.3.1 Adjusting the supply voltage (VDDE)

The VDDE voltage is directly used as reference and source to drive the LCD segments. The contrast could be adjusted by altering the voltage VDDE, but be aware that this voltage is used for the padring also.

56.6.3.2 Adding Contrast Adjustment Phases

Another way to adjust the contrast is to add another phase to the refresh cycle which keeps the voltage the same on all frontplane and backplane electrodes (V_{SS} in this case). This will contribute towards lowering the V_{ONRMS} and V_{OFFRMS} voltages over the segment. For this reason, the whole frame is divided into steps. The value from the Contrast Control register (LCC) determines how many of these steps are taken by the contrast phase. Whenever the length of the contrast adjustment phase is changed, the length of the other phases is automatically changed to ensure the frame length remains the same.

Example: A value of 256 in the Contrast Control (LCC) register will force the contrast phase to take approximately a 256/2047 of the length of the whole frame.

The figure below shows an example of waveform with 1/1 duty and 1/1 bias ratios with the additional contrast adjustment phases

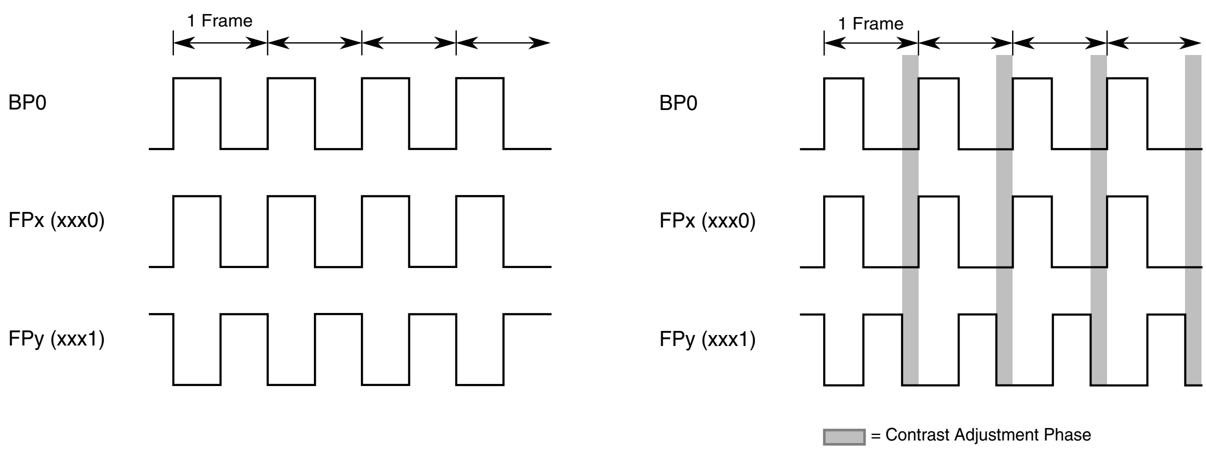


Figure 56-17. Contrast Adjustment Phases

Note

RMS stands for root mean square and is a statistical measure of the magnitude of a varying quantity. It is calculated with the formula:

$$V_{\mathrm{RMS}} = \sqrt{\frac{\int_0^T V^2 \mathrm{dt}}{T}}$$

56.6.4 LCD RAM

For a segment on the LCD to be displayed, data must be written to the LCD RAM which is shown in Memory Map and Register Definition. The bits in the LCD RAM correspond to the segments that are driven by the frontplane and backplane drivers. Writing a '1' to a given location will result in the corresponding display segment being driven with a differential RMS voltage, based on selected reference, necessary to turn the segment ON when the LCDEN bit is set and the corresponding ENFP[*n*-1:0] bit is set. Writing a '0' to a given location will result in the corresponding display segment being driven with a differential RMS voltage necessary to turn the segment OFF. The LCD RAM is a dual port RAM that interfaces with the internal address and data buses of the MCU. It is possible to read from LCD RAM locations for scrolling purposes. Writing or reading of the LCDEN bit does not change the contents of the LCD RAM. After a reset, the LCD RAM contents will be cleared.

56.6.5 LCD Driver System Enable and Frontplane Enable Sequencing

If LCDEN = 0 (LCD Driver System disabled) and the frontplane enable bit, ENFP[*n*-1:0], is set, the frontplane driver waveform will not appear on the output until LCDEN is set. If LCDEN = 1 (LCD Driver System enabled), the frontplane driver waveform will appear on the output as soon as the corresponding frontplane enable bit, ENFP[*n*-1:0], in the registers ENFPR0 and ENFPR1 is set.

56.6.6 LCD Driver Backplane Remapping

The backplane and frontplane pins can be remapped/swapped using LCDBPS and LCDBPA. While LCDBPA adds the non available backplane waveforms onto FP pins, LCDBPS does a swapping of backplanes waveforms with frontplanes waveforms on selected pins. The swapping of LCDBPS depends on the availability of Frontplane pins FP[n-1:0]. If the number of frontplanes n implemented is not sufficient no remapping will occur. See the following table for details.

Table 56-22. Backplane Remapping

LCDBPS	Condition: FP[n-1:0]	Remapping LCDBPA[1:0]=00 or LCDBPA[1:0]=11 BP[m-1:0]	Remapping LCDBPA[1:0]=01BP[m-1:0]	Remapping LCDBPA[1:0]=10BP[m-1:0]
000		no remapping	FP[5-m:0] <- BP[5:m] if m=6 no remapping	FP[7-m:0] <- BP[7:m] if m=8 no remapping
001		BP[m-1:0] <- FP[m-1+4:4] FP[m-1+4:4] <- BP[m-1:0]	BP[m-1:0] <- FP[m-1+4:4] FP[5+4:4] <- BP[5:0]	BP[m-1:0] <- FP[m-1+4:4] FP[7+4:4] <- BP[7:0]
010		BP[m-1:0] <- FP[m-1+12:12] FP[m-1+12:12] <- BP[m-1:0]	BP[m-1:0] <- FP[m-1+12:12] FP[5+12:12] <- BP[5:0]	BP[m-1:0] <- FP[m-1+12:12] FP[7+12:12] <- BP[7:0]
011		BP[m-1:0] <- FP[m-1+20:20] FP[m-1+20:20] <- BP[m-1:0]	BP[m-1:0] <- FP[m-1+20:20] FP[5+20:20] <- BP[5:0]	BP[m-1:0] <- FP[m-1+20:20] FP[7+20:20] <- BP[7:0]
100	n=36 or n>36	BP[m-1:0] <- FP[m-1+28:28] FP[m-1+28:28] <- BP[m-1:0]	BP[m-1:0] <- FP[m-1+28:28] FP[5+28:28] <- BP[5:0]	BP[m-1:0] <- FP[m-1+28:28] FP[7+28:28] <- BP[7:0]
101	n=44 or n>44	BP[m-1:0] <- FP[m-1+36:36] FP[m-1+36:36] <- BP[m-1:0]	BP[m-1:0] <- FP[m-1+36:36] FP[5+36:36] <- BP[5:0]	BP[m-1:0] <- FP[m-1+36:36] FP[7+36:36] <- BP[7:0]
110	n=52 or n>52	BP[m-1:0] <- FP[m-1+44:44] FP[m-1+44:44] <- BP[m-1:0]	BP[m-1:0] <- FP[m-1+44:44] FP[5+44:44] <- BP[5:0]	BP[m-1:0] <- FP[m-1+44:44] FP[7+44:44] <- BP[7:0]
111	n=60 or n>60	BP[m-1:0] <- FP[m-1+52:52] FP[m-1+52:52] <- BP[m-1:0]	BP[m-1:0] <- FP[m-1+52:52] FP[5+52:52] <- BP[5:0]	BP[m-1:0] <- FP[m-1+52:52] FP[7+52:52] <- BP[7:0]

Examples:

1. 40 (n=40) Frontplanes and 4 (m=4) backplanes are implemented. LCDBPS is set to 000 and LCDBPA is set to 01

   Frontplanes FP[39:2] and BP[3:0] will stay the same, but FP[1:0] pins will controlled by BP[5:4] functionality.

2. 40 (n=40) Frontplanes and 4 (m=4) backplanes are implemented. LCDBPS is set to 010 and LCDBPA is set to 00

   Frontplanes FP[39:16] and FP[11:0] will stay the same, but FP[15:12] pins are swapped with BP[3:0].

3. 40 (n=40) Frontplanes and 4 (m=4) backplanes are implemented. LCDBPS is set to 010 and LCDBPA is set to 01

   Frontplanes FP[39:18] and FP[11:0] will stay the same. FP[15:12] pins are swapped with BP[3:0]. FP[17:16] is replaced by BP[5:4] functionality.

4. 40 (n=40) Frontplanes and 4 (m=4) backplanes are implemented. LCDBPS is set to 101 and LCDBPA is set to 01

No remapping: Frontplanes FP[39:0] and BP[3:0] will stay the same, because condition n>=44 is not fulfilled.

56.6.7 LCD Bias and Modes of Operation

The LCD64F6B driver has seven modes of operation:

- 1/1 Duty (1 backplane), 1/1 Bias (2 voltage levels)
- 1/2 Duty (2 backplanes), 1/2 Bias (3 voltage levels)
- 1/2 Duty (2 backplanes), 1/3 Bias (4 voltage levels)
- 1/3 Duty (3 backplanes), 1/3 Bias (4 voltage levels)
- 1/4 Duty (4 backplanes), 1/3 Bias (4 voltage levels)
- 1/5 Duty (5 backplanes), 1/3 Bias (4 voltage levels)
- 1/6 Duty (6 backplanes), 1/3 Bias (4 voltage levels)
- 1/8 Duty (8 backplanes), 1/3 Bias (4 voltage levels)

The voltage levels required for the different operating modes are generated internally based on selected reference voltage. VDDE as used as reference and supply. Changing VDDE alters the differential RMS voltage across the segments in the ON and OFF states, thereby can change the display contrast.

The backplane waveforms are continuous and repetitive every frame. They are fixed within each operating mode and are not affected by the data in the LCD RAM.

The frontplane waveforms generated are dependent on the state (ON or OFF) of the LCD segments as defined in the LCD RAM. The LCD driver hardware uses the data in the LCD RAM to construct the frontplane waveform to create a differential RMS voltage necessary to turn the segment ON or OFF.

The LCD duty is decided by the DUTY bits in the LCD Control Register (LCDCR). The number of bias voltage levels is determined by the BIAS bit in the LCDCR. The following table summarizes the Multiplex modes (duties) and the bias voltage levels that can be selected for each multiplex mode (duty). The backplane pins have their corresponding backplane waveform output BP[*m*-1:0] in high impedance state when in the OFF state as indicated in the following table. In the OFF state the corresponding pins BP[*m*-1:0]can be used for other functionality.

Table 56-23. LCD Duty and Bias

<table>
<tr><th></th><th>LCDCR Register</th><th colspan="8">Backplanes</th><th colspan="2">Bias Level</th></tr>
<tr><th>Dut y</th><th>DUTY</th><th>BP7</th><th>BP6</th><th>BP5</th><th>BP4</th><th>BP3</th><th>BP2</th><th>BP1</th><th>BP0</th><th>BIAS=0</th><th>BIAS=1</th></tr>
<tr><td>1/1</td><td>000</td><td colspan="7">OFF</td><td>ON</td><td colspan="2">1/1</td></tr>
<tr><td>1/2</td><td>001</td><td colspan="6">OFF</td><td colspan="2">ON</td><td>1/2</td><td>1/3</td></tr>
<tr><td>1/3</td><td>010</td><td colspan="5">OFF</td><td colspan="3">ON</td><td colspan="2">1/3</td></tr>
<tr><td>1/4</td><td>011</td><td colspan="4">OFF</td><td colspan="4">ON</td><td colspan="2">1/3</td></tr>
<tr><td>1/5</td><td>100</td><td colspan="3">OFF</td><td colspan="5">ON</td><td colspan="2">1/3</td></tr>
<tr><td>1/6</td><td>101</td><td colspan="2">OFF</td><td colspan="6">ON</td><td colspan="2">1/3</td></tr>
<tr><td>1/6</td><td>110</td><td colspan="2">OFF</td><td colspan="6">ON</td><td colspan="2">1/3</td></tr>
<tr><td>1/8</td><td>111</td><td colspan="8">ON</td><td colspan="2">1/3</td></tr>
</table>

56.6.8 Operation in Power Saving Modes

56.6.8.1 Operation in STOP mode

The LCD64F6B system operation during STOP mode is controlled by the LCDRST bit in the LCD Control Register (LCDCR).

If the LCD64F6B is requested to enter STOP mode and LCDRST is cleared while LCDEN is set the LCD waveform generation clocks are stopped and the LCD64F6B drivers pull down to ground those frontplane and backplane pins that were enabled before entering STOP mode.The contents of the LCD RAM and the LCD registers retain the values they had prior to entering stop mode.

If LCDRST is set while LCDEN is set and the LCD64F6B is requested to enter STOP mode the LCD waveform generation is continued.

56.6.8.2 Operation in LPSTOP Mode

The LCD64F6B system operation during LPSTOP Modemode is controlled by the LCDRST bit in the LCD Control Register (LCDCR).

If the LCD64F6B is not powered down by the system, and is requested to enter LPSTOP Modemode and LCDRST is cleared while LCDEN is set the LCD waveform generation clocks are stopped and the LCD64F6B drivers pull down to ground those frontplane and

backplane pins that were enabled before entering LPSTOP Modemode.The contents of the LCD RAM and the LCD registers retain the values they had prior to entering LPSTOP Modemode.

If the LCD64F6B is not powered down by the system, and is requested to enter LPSTOP Modemode and LCDRST is set while LCDEN is set the LCD waveform generation is continued.

Note

The user needs to take care that the system keeps the clock applied to the running LCD64F6B module during all modes where it is enabled. If no clock is applied while the LCD64F6B module is running the LCD could be damaged.

56.6.9 Other Power Saving

The LCD64F6B has features to adjust the frontplane & backplane drive strength and to boost the drive strength while the planes are switching.

56.6.9.1 LCD Reference Clock Select

Using LCDRCS the LCD reference clock can be selected. If LCDRCS is cleared, the system clock is applied as reference clock to the prescaler input. If LCDRCS is set, the source clock for LCD Driver system (prescaler input) is SXOSC clock. Selecting the lower power consuming clock of both as reference clock for the prescaler input can save power consumption.

56.6.9.2 Boost at Switching

Since the LCD appears like a capacitance to the driver it is sometimes useful to boost the output current during transitions to increase the slew rate of the driver. If boosting is enabled, the drive strength capability is increased a little ahead in time when the planes are switching, and reduced again to normal drive after a certain time. To enable the boosting BSTEN need to be set. If BSTEN is not set, standard drive strength is used.

Using the BST bit if BSTEN is set, LCD output current magnification (boost) can be selected. Available is 8 times boosting, when BST is deasserted and 16 times boosting when BST is asserted.

The selected boost via BST bit can be switched to be always on by setting BSTAO, if BSTEN is set.

56.6.9.3 Standard Drive Selection

The output current has a direct impact on the power consumption and drive capability of the LCD64F6B module. The PWR bits select the output current height and can be used to influence power consumption and drive strength. The following table lists the possible selections of PWR.

Table 56-24. Output Current Selection (PWR)

PWR	Current
00	Standard Current
01	2 * Standard Current
10	3 * Standard Current
11	4 * Standard Current

56.6.9.4 Usage Recommendation

It is recommended to start with maximum output current and maximum enabled boosting. As long as the LCD is fully functional in all environments a reduction of Output Current and boosting can be done.

When reducing the Output Current (PWR), the power consumption of the LCD module and the maximum switching current of the planes is reduced.

When reducing or disabling the boosting (BST,BSTEN), the output current, while planes are switching, is reduced.

The user can select any combination of PWR, BST, BSTEN, BSTAO to what fit his needs best. The following figure gives an overview what is the impact when taking advance of Boost at Switching and Standard Drive Selection.

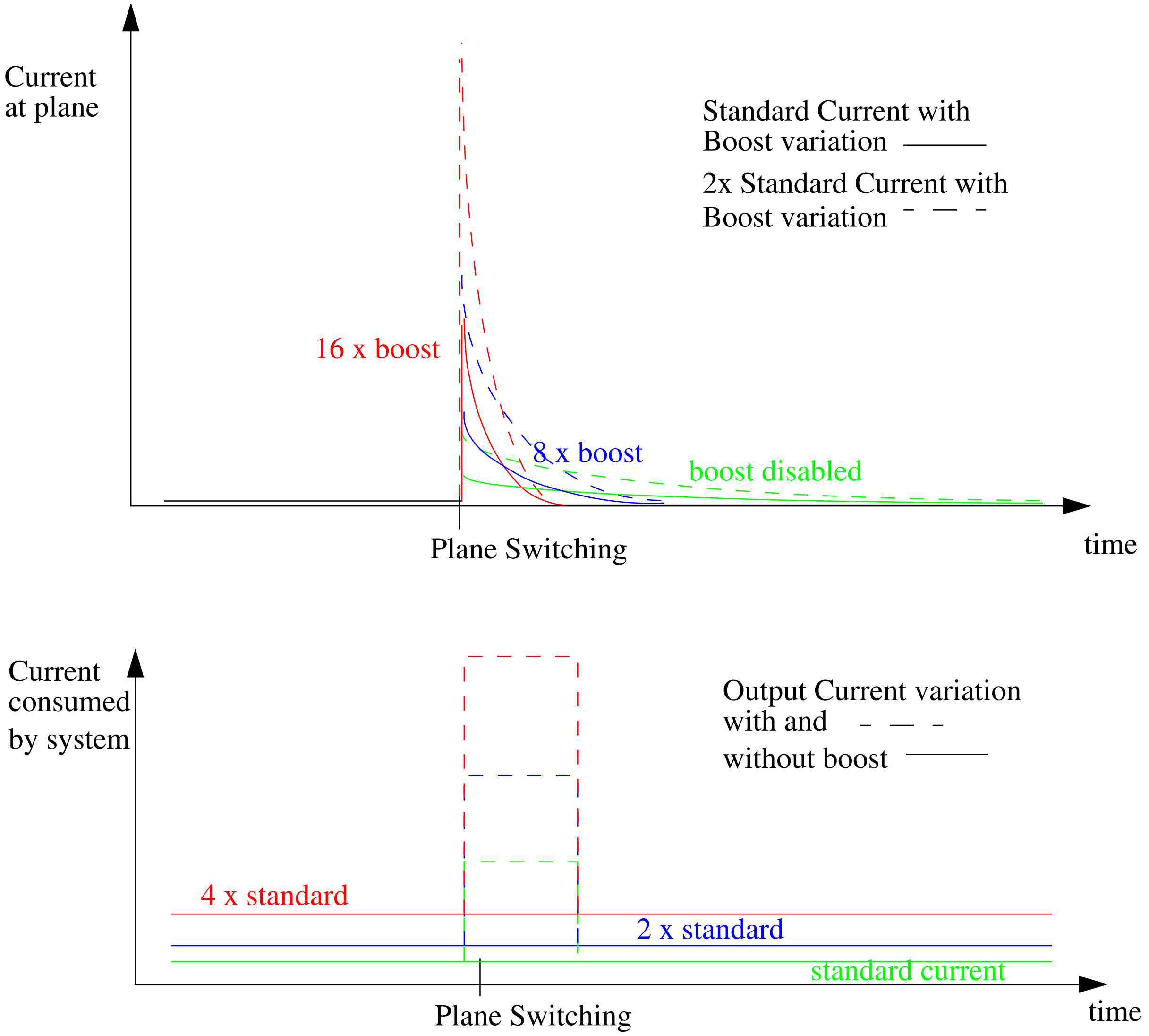


Figure 56-18. PWR, BST, BSTEN Example Diagram

56.6.10 Interrupts

This section describes all interrupts originated by LCD64F6B.

56.6.10.1 EOF Interrupt

LCDEN must be set to enable the interrupt at end of frame feature. Every time a frame generation was completed a counter is decremented by 1. If the counter reaches zero End of Frame interrupt Flag (EOF) bit is set and the counter is reset to NOF. The counter is reset to request interrupt at the end of every frame, as if NOF would be 0x00, if LCDEN is asserted while the LCD is off.

The EOF flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LCDINT = 1), EOF causes an interrupt request. The number of frames (NOF) bits determine how many frames are count until EOF flag is set. The possible number of frames are:

NOF= 0x00: An interrupt is requested at the end of every frame.

NOF= 0x01: An interrupt is requested at the end of every second frame.

NOF= 0x02: An interrupt is requested at the end of every third frame.

…

NOF= 0xFF: An interrupt is requested at the end of every 256th frame.

56.7 LCD Waveform Examples

The following figures show the timing examples of the LCD output waveforms for the available modes of operation. The contrast control is disabled in all examples (CCEN = 0 in LCDCCR). Reference voltage is VDDE.

56.7.1 1/1 Duty Multiplexed with 1/1 Bias Mode

Duty = 1/1:DUTY = 000

Bias = 1/1:BIAS = 0 or BIAS = 1

$V_0 = V_1 =$ VSSE $V_2 = V_3 =$ VDDE

- Only BP0 is used, a maximum of 64 segments are displayed.

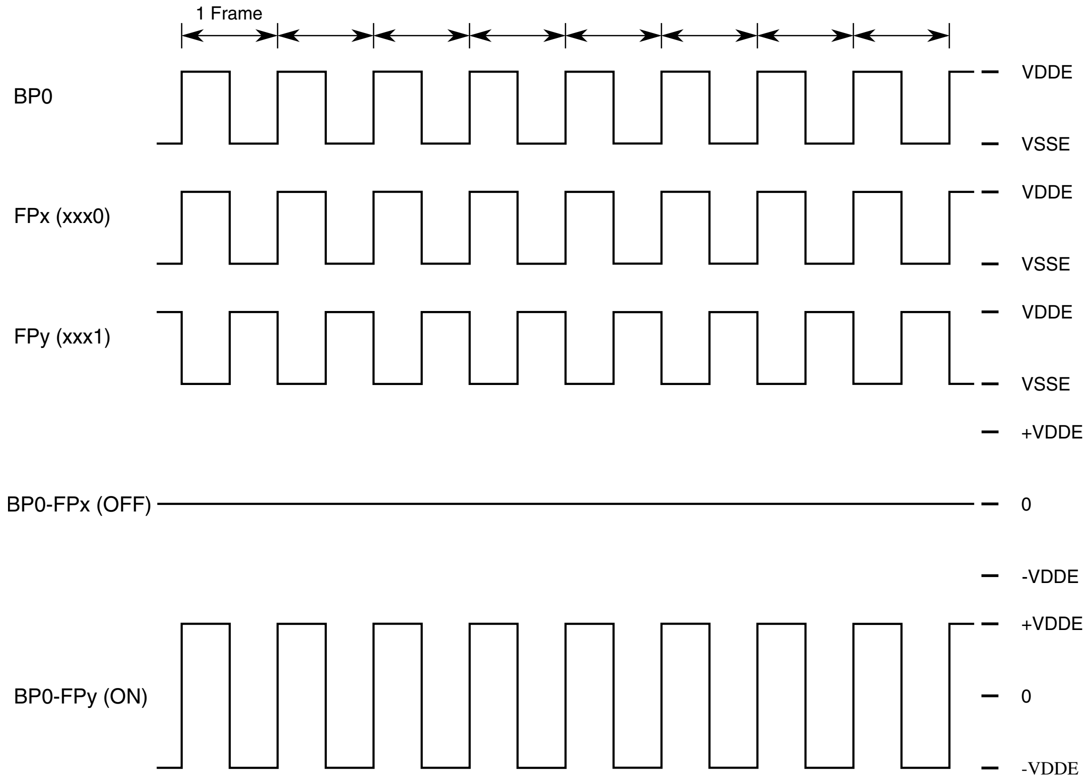


Figure 56-19. 1/1 Duty and 1/1 Bias

56.7.2 1/2 Duty Multiplexed with 1/2 Bias Mode

Duty = 1/2:DUTY = 001

Bias = 1/2:BIAS = 0

V_0 = VSSE, V_1 = VDDE * 1/3, V_2 = VDDE * 2/3, V_3 = VDDE

- Only BP0 and BP1 are used, a maximum of 128 segments are displayed.

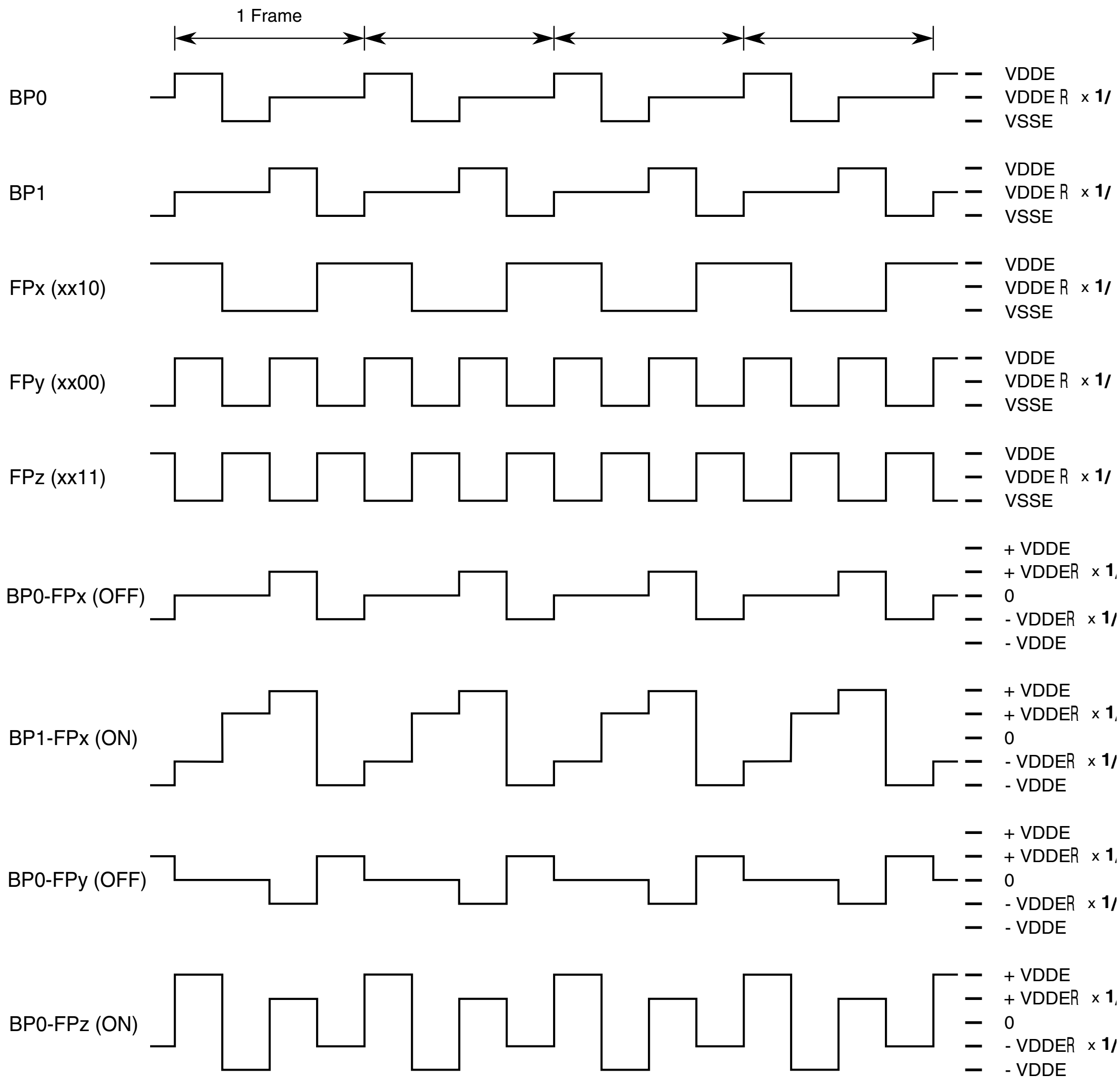


Figure 56-20. 1/2 Duty and 1/2 Bias

56.7.3 1/2 Duty Multiplexed with 1/3 Bias Mode

Duty = 1/2:DUTY = 001

Bias = 1/3:BIAS = 1

V_0 = VSSE, V_1 = VDDE * 1/3, V_2 = VDDE * 2/3, V_3 = VDDE

- Only BP0 and BP1 are used, a maximum of 128 segments are displayed.

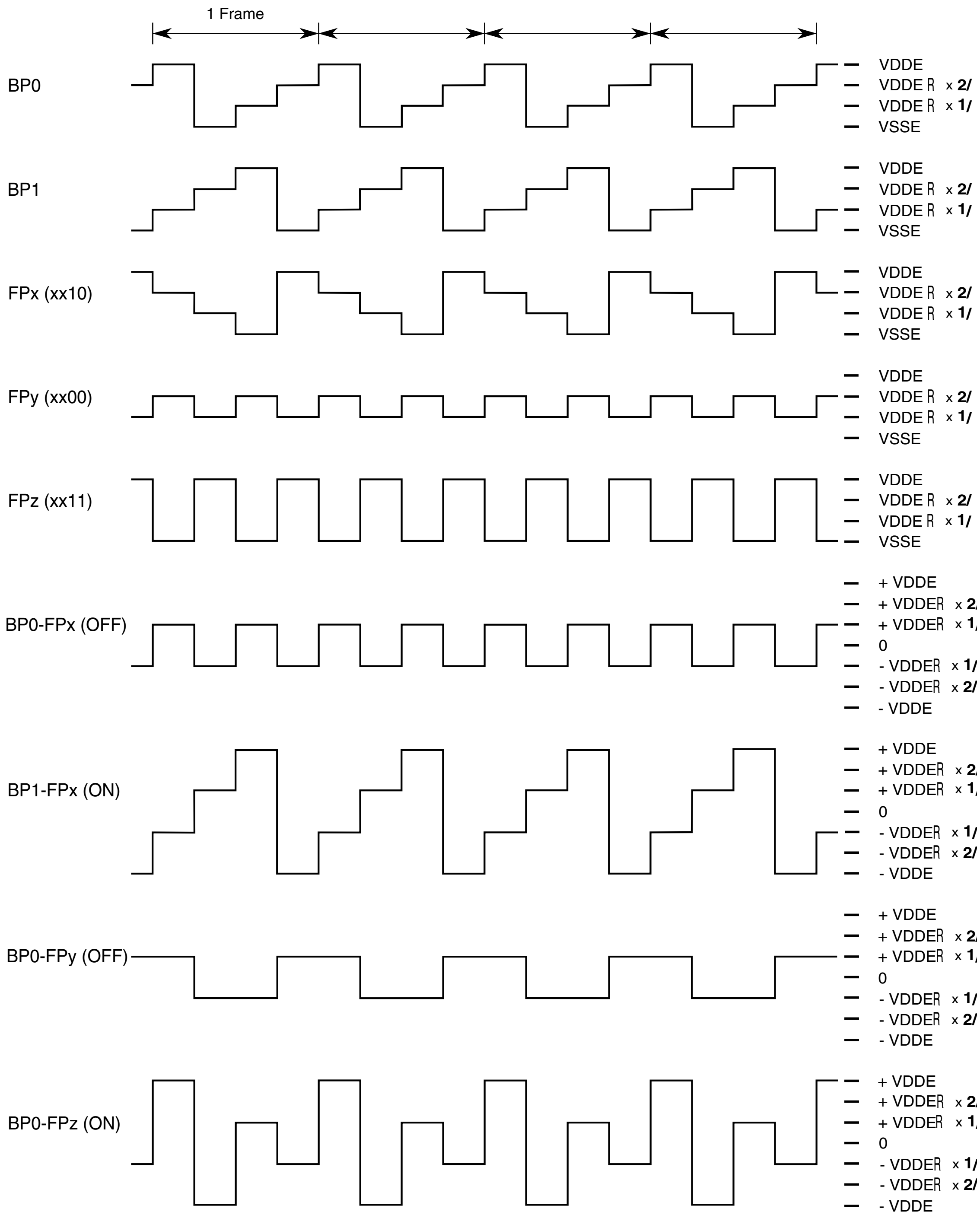


Figure 56-21. 1/2 Duty and 1/3 Bias

56.7.4 1/3 Duty multiplexed with 1/3 Bias mode

Duty = 1/3:DUTY = 010

Bias = 1/3:BIAS = 0 or BIAS = 1

V_0 = VSSE, V_1 = VDDE * 1/3, V_2 = VDDE * 2/3, V_3 = VDDE

- Only BP0, BP1 and BP2 are used, a maximum of 192 segments are displayed.

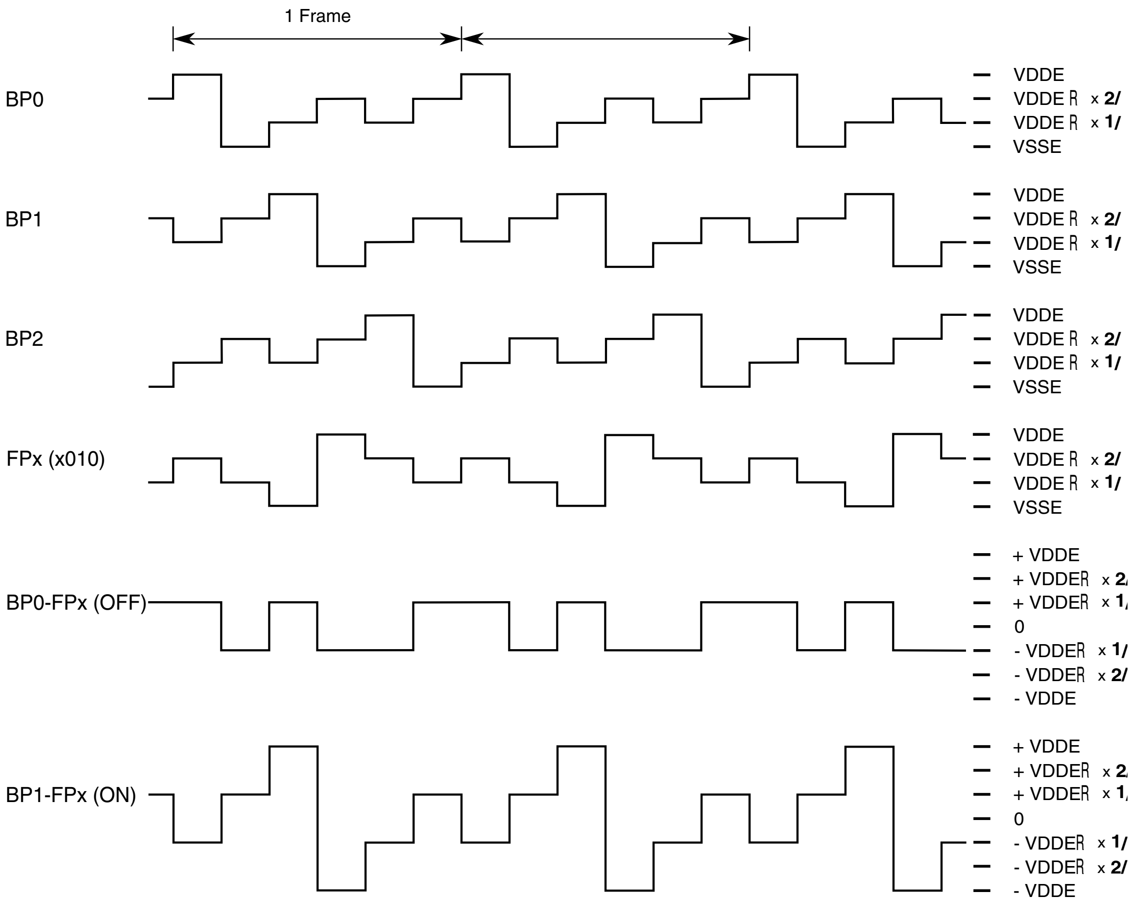


Figure 56-22. 1/3 Duty and 1/3 Bias

56.7.5 1/4 Duty multiplexed with 1/3 Bias mode

Duty = 1/4:DUTY = 011

Bias = 1/3:BIAS = 0 or BIAS = 1

V_0 = VSSE, V_1 = VDDE * 1/3, V_2 = VDDE * 2/3, V_3 = VDDE

- BP4 and BP5 are not used, a maximum of 256 segments are displayed.

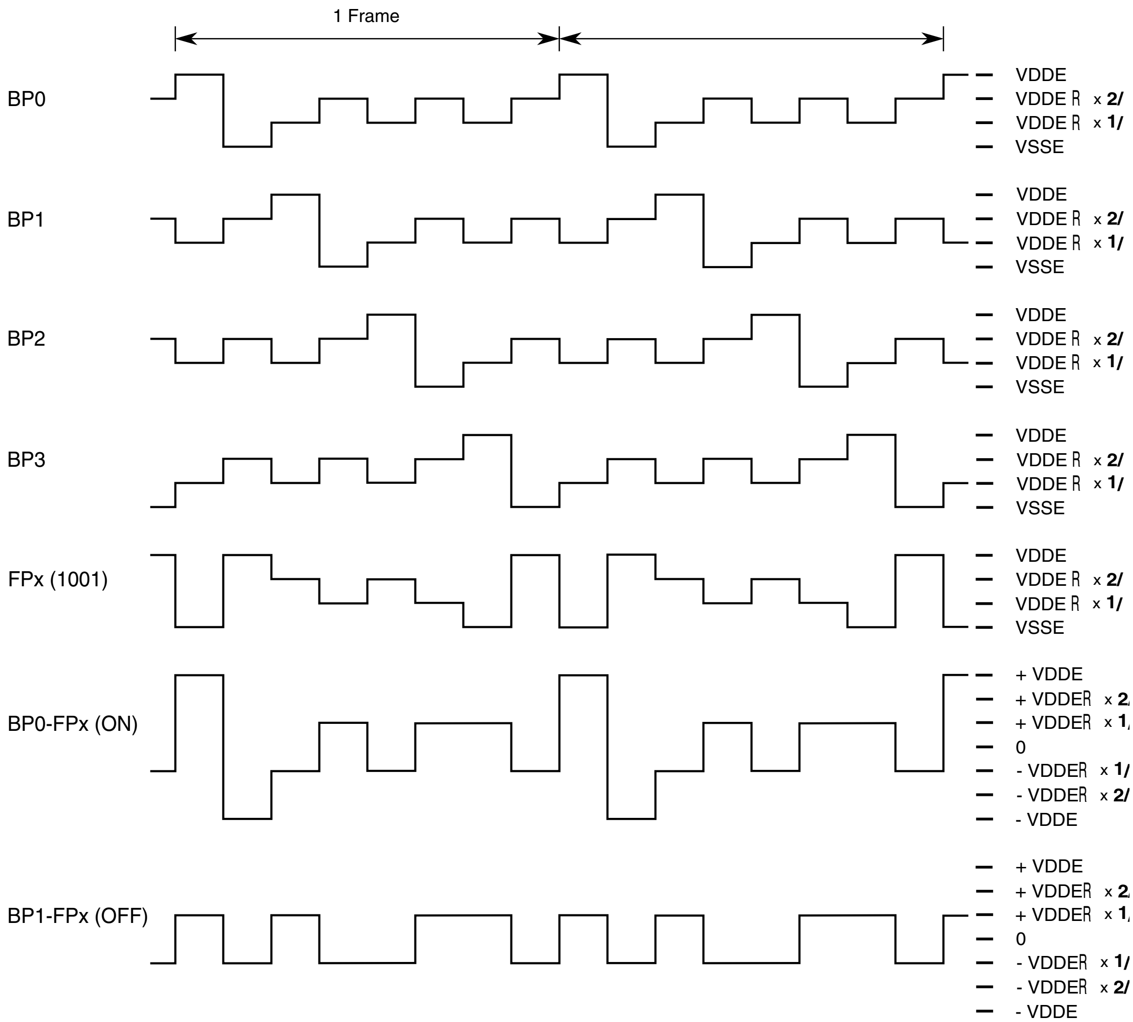


Figure 56-23. 1/4 Duty and 1/3 Bias

56.7.6 1/5 Duty multiplexed with 1/3 Bias

Duty = 1/5:DUTY = 100

Bias = 1/3:BIAS = 0 or BIAS = 1

V_0 = VSSE, V_1 = VDDE * 1/3, V_2 = VDDE * 2/3, V_3 = VDDE

- BP5 is not used, a maximum of 320 segments are displayed.

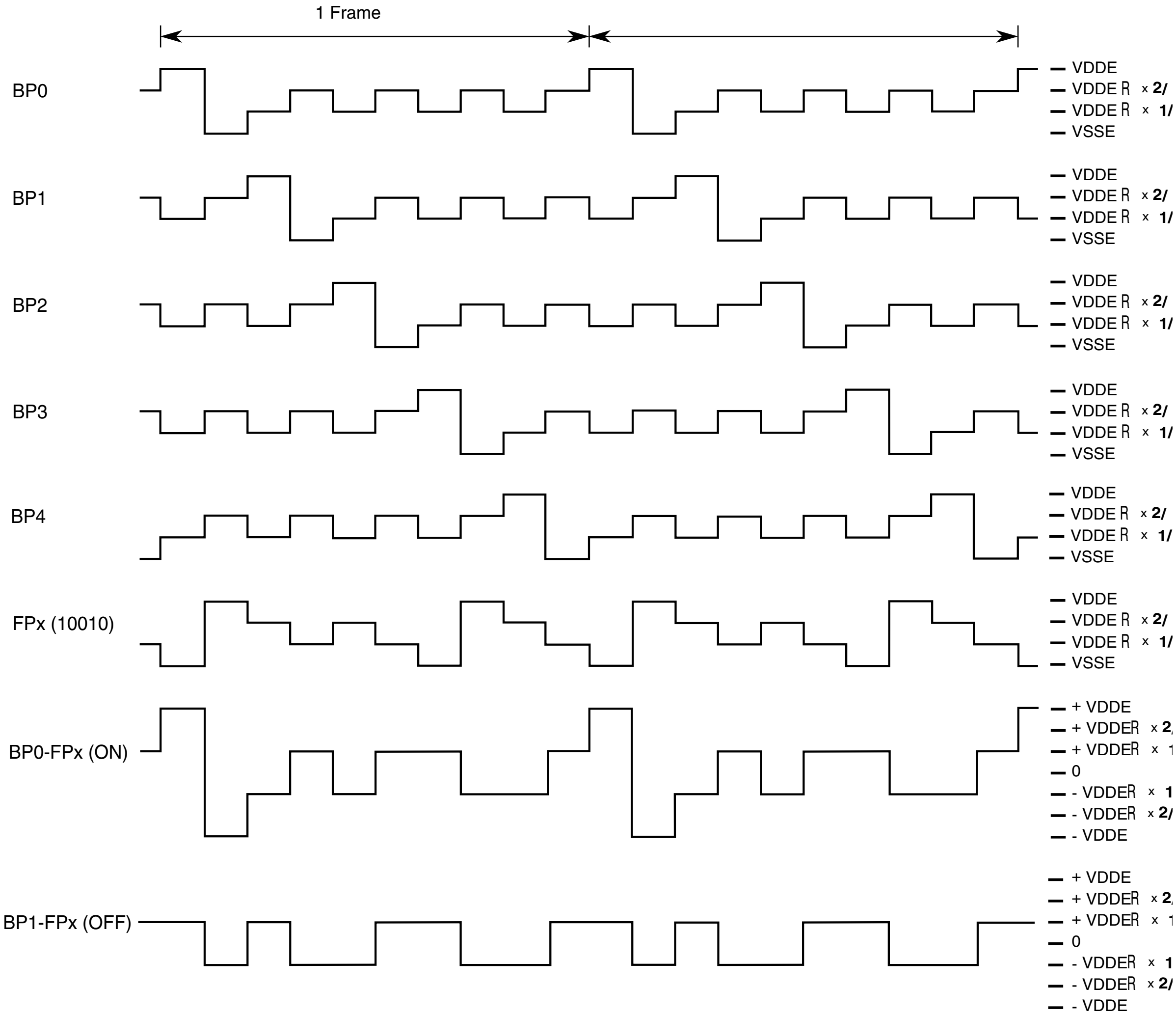


Figure 56-24. 1/5 Duty and 1/3 Bias

56.7.7 1/6 Duty multiplexed with 1/3 Bias mode

Duty = 1/5:DUTY = 101

Bias = 1/3:BIAS = 0 or BIAS = 1

V_0 = VSSE, V_1 = VDDE * 1/3, V_2 = VDDE * 2/3, V_3 = VDDE

- All backplanes are used, a maximum of 384 segments are displayed.

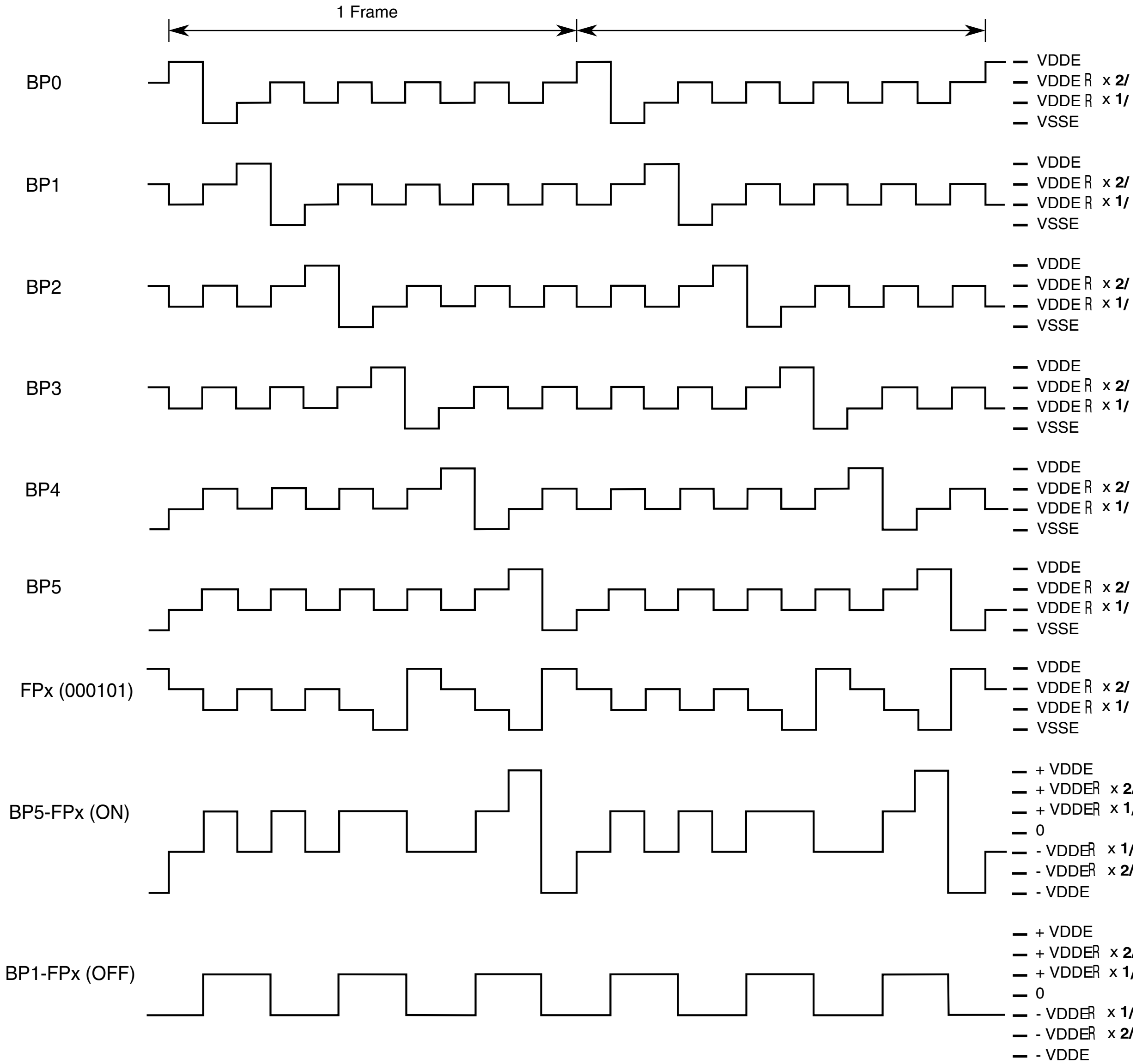


Figure 56-25. 1/6 Duty and 1/3 Bias

56.8 Initialization Information

This is a step-wise example instruction for initializing. The initial values of all registers are the reset values.

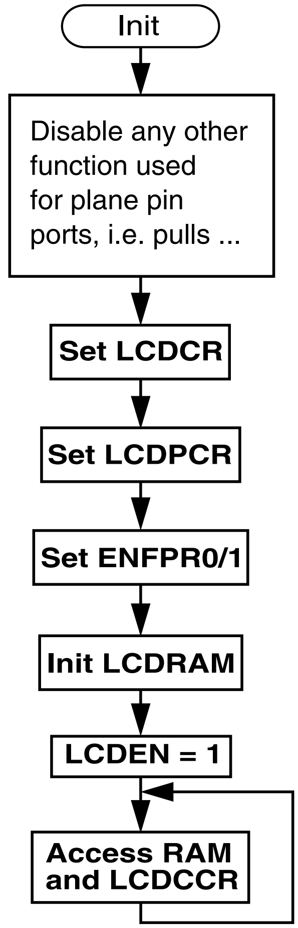


Figure 56-26. Example Initialization Diagram

Chapter 57
Video-In VIU3

57.1 Block Diagram

The figure below shows a block diagram of the Video-In 3 (VIU3).

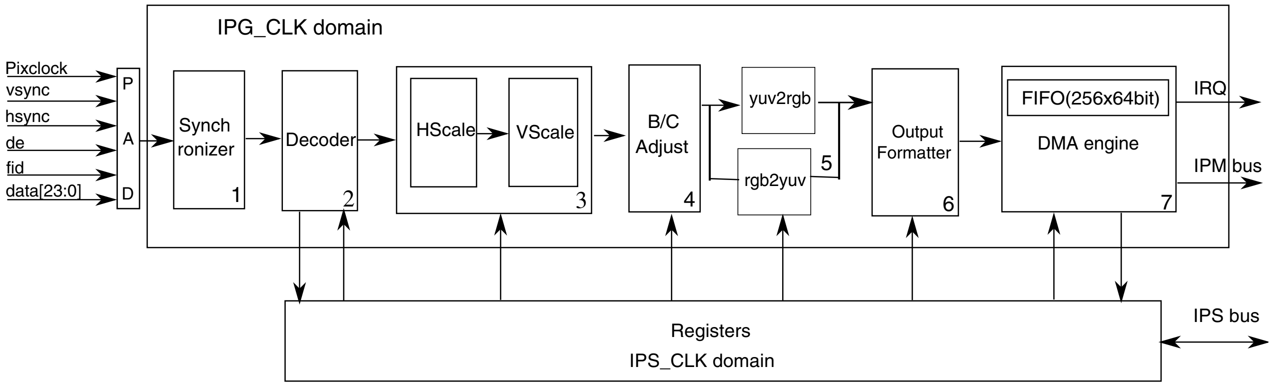


Figure 57-1. VIU3 Block Diagram

57.2 Features

- Supports from QVGA to XVGA input/output resolution
- 8-bit ITU656 video input[1]
- RGB888/RGB666/RGB565 parallel input
- RGB888 input with one color component transferred per clock
- 24-bit YUV parallel input from analog video decoder
- Up to 1/8 video down-scaling on horizontal and vertical direction
- Up to 2/1 horizontal video up-scaling
- Horizontal mirroring
- Brightness/contrast adjust
- YUV to RGB 888/565 conversion
- RGB to YUV conversion when input is RGB

1. When down scaling and/or B/C adjust is enabled, the two LSB's of the 10-bit input are ignored.

- Simple de-interlace function (weaving) for interlaced or pseudo interlaced video input
- Internal DMA engine for transferring data from FIFO to system memory

57.3 Video Input Signal Mapping

Table 57-1. Video Input Signals

Signal	RGB565	RGB666	RGB888	RGB888 (3 cycle)	ITU656 (10-bit)	ITU656 (8-bit)	Parallel YUV (24-bit)
ipp_pix_data[23]	R4	R5	R7	—	—	—	Y7
ipp_pix_data[22]	R3	R4	R6	—	—	—	Y6
ipp_pix_data[21]	R2	R3	R5	—	—	—	Y5
ipp_pix_data[20]	R1	R2	R4	—	—	—	Y4
ipp_pix_data[19]	R0	R1	R3	—	—	—	Y3
ipp_pix_data[18]	0	R0	R2	—	—	—	Y2
ipp_pix_data[17]	0	0	R1	—	—	—	Y1
ipp_pix_data[16]	0	0	R0	—	—	—	Y0
ipp_pix_data[15]	G5	G5	G7	—	—	—	U7
ipp_pix_data[14]	G4	G4	G6	—	—	—	U6
ipp_pix_data[13]	G3	G3	G5	—	—	—	U5
ipp_pix_data[12]	G2	G2	G4	—	—	—	U4
ipp_pix_data[11]	G1	G1	G3	—	—	—	U3
ipp_pix_data[10]	G0	G0	G2	—	—	—	U2
ipp_pix_data[9]	0	0	G1	—	Y9/C9	Y7/C7	U1
ipp_pix_data[8]	0	0	G0	—	Y8/C8	Y6/C6	U0
ipp_pix_data[7]	B4	B5	B7	R7/G7/B7	Y7/C7	Y5/C5	V7
ipp_pix_data[6]	B3	B4	B6	R6/G6/B6	Y6/C6	Y4/C4	V6
ipp_pix_data[5]	B2	B3	B5	R5/G5/C5	Y5/C5	Y3/C3	V5
ipp_pix_data[4]	B1	B2	B4	R4/G4/C4	Y4/C4	Y2/C2	V4
ipp_pix_data[3]	B0	B1	B3	R3/G3/C3	Y3/C3	Y1/C1	V3
ipp_pix_data[2]	0	B0	B2	R2/G2/C2	Y2/C2	Y0/C0	V2
ipp_pix_data[1]	0	0	B1	R1/G1/C1	Y1/C1	0	V1
ipp_pix_data[0]	0	0	B0	R0/G0/C0	Y0/C0	0	V0
ipp_ind_pix_clk	CLK	CLK	CLK	CLK	CLK	CLK	CLK
ipp_ind_pix_hs	HSYNC	HSYNC	HSYNC	HSYNC	—	—	HSYNC
ipp_ind_pix_vs	VSYNC	VSYNC	VSYNC	VSYNC	—	—	VSYNC
ipp_ind_pix_de	DE	DE	DE	DE	—	—	DE
ipp_ind_pix_fid	FID	FID	FID	FID	—	—	FID

57.4 Memory map and register definition

The memory map for the VIU3 module is given below. The total address for each register is the sum of the base address for the VIU3 module and the address offset for each register. Also, please note that the bit order for each register bit field is [highest:lowest], whereas the overall register bits are numbered [lowest:highest].

VIU3 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_9000	Status And Configuration Register (VIU3_SCR)	32	R/W	0280_0000h	57.4.1/3264
400C_9004	Luminance Coefficients For Red, Green And Blue Matrix (VIU3_LUMA_COMP)	32	R/W	9512_A254h	57.4.2/3267
400C_9008	Chroma Coefficients For Red Matrix (VIU3_CHROMA_RED)	32	R/W	0331_0000h	57.4.3/3268
400C_900C	Chroma Coefficients For Green Matrix (VIU3_CHROMA_GREEN)	32	R/W	0660_0F38h	57.4.4/3269
400C_9010	Chroma Coefficients For Blue Matrix (VIU3_CHROMA_BLUE)	32	R/W	0000_0409h	57.4.5/3269
400C_9014	Base Address Of Every Field/Frame Of Picture In Memory (VIU3_DMA_ADDR)	32	R/W	0000_0000h	57.4.6/3270
400C_9018	Horizontal DMA Increment (VIU3_DMA_INC)	32	R/W	0000_0000h	57.4.7/3270
400C_901C	Input Video Pixel and Line Count (VIU3_INVSZ)	32	R/W	00F0_02D0h	57.4.8/3271
400C_9020	High IPM Request Priority Alarm (VIU3_HPRALRM)	32	R/W	0000_0090h	57.4.9/3271
400C_9024	Programable Alpha Value (VIU3_ALPHA)	32	R/W	0000_00FFh	57.4.10/ 3272
400C_9028	Down Scaling Factor In Horizontal Direction (VIU3_HFACTOR)	32	R/W	0000_0100h	57.4.11/ 3272
400C_902C	Down Scaling Factor In Vertical Direction (VIU3_VFACTOR)	32	R/W	0000_0100h	57.4.12/ 3273
400C_9030	Down Scaling Destination Pixel and Line Count (VIU3_VID_SIZE)	32	R/W	00F0_02D0h	57.4.13/ 3273
400C_9034	B/C Adjust Look-up-table Current Address (VIU3_LUT_ADDR)	32	R/W	0000_0000h	57.4.14/ 3274
400C_9038	B/C Adjust Look-up-table Data Entry (VIU3_LUT_DATA)	32	R/W	0000_0000h	57.4.15/ 3274
400C_903C	Extended Configuration Register (VIU3_EXT_CONFIG)	32	R/W	0000_0100h	57.4.16/ 3275
400C_9040	Red, Green and Blue Coefficients for Luminance component (VIU3_RGB_Y)	32	R/W	2108_1032h	57.4.17/ 3276
400C_9044	Red, Green and Blue Coefficients for Chroma U component (VIU3_RGB_U)	32	R/W	1304_A8E1h	57.4.18/ 3277
400C_9048	Red, Green and Blue Coefficients for Chroma V component (VIU3_RGB_V)	32	R/W	3845_E024h	57.4.19/ 3278

57.4.1 Status And Configuration Register (VIU3_SCR)

Address: 400C_9000h base + 0h offset = 400C_9000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MODE32BIT	ROUND_ON	DITHER_ON	FIELD_NO	DMA_ACT	SCALER_EN	YUV2RGB_EN	BC_EN	MODE444	Reserved	ERROR_IRQ	DMA_END_IRQ	VSTART_IRQ	HSYNC_IRQ	VSYNC_IRQ	FIELD_IRQ
W											w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	ECC_EN	ERROR_EN	DMA_END_EN	VSTART_EN	HSYNC_EN	VSYNC_EN	FIELD_EN	ERROR_CODE				FORMAT_CTRL			SOFT_RESET
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VIU3_SCR field descriptions

Field	Description
31 MODE32BIT	Select 32-bit or 16-bit output from the output formatter (block 6 in) **NOTE:** For RGB666 parallel input, when MODE32BIT is cleared, the output is {R[5:1], G[5:0], B[5:1]}; when MODE32BIT is set, the output is {R[5:0], 2'b00, G[5:0], 2'b00, B[5:0], 2'b00} 0 16-bit RGB or YUV 4:2:2 output 1 32-bit RGB or YUV 4:4:4 output. DITHER_ON and ROUND_ON are ignored if output is 32-bit RGB.
30 ROUND_ON	Round is on. Used when video data is stored in buffer as RGB565 format.
29 DITHER_ON	Dithering is on. Used when video data is stored in buffer as RGB565 format and ROUND_ON is not set.
28 FIELD_NO	Field number, extracted from ITU-656 stream.
27 DMA_ACT	DMA transfer of current field/frame is busy (write by software, cleared at end of transfer). When DMA_ACT is cleared, input video data is ignored and not put into FIFO.
26 SCALER_EN	Down scaling enable.
25 YUV2RGB_EN	YUV to RGB conversion enable.
24 BC_EN	Bright/Contrast adjust enable.
23 MODE444	YUV 4:4:4 mode enable bit. When it is set ITU decoder sends out YUV 4:4:4 format data, otherwise YUV 4:2:2 is sent by default. Down scaler works on YUV 4:2:2 format. So this bit shall be cleared when scaling is enabled.
22 RESERVED	This field is reserved.

Table continues on the next page...

VIU3_SCR field descriptions (continued)

Field	Description
21 ERROR_IRQ	Interrupt status bit. Write '1' to clear ERROR_IRQ.
20 DMA_END_IRQ	Interrupt status bit. Write '1' to clear DMA_END_IRQ.
19 VSTART_IRQ	Interrupt status bit. Write '1' to clear VSTART_IRQ.
18 HSYNC_IRQ	Interrupt status bit. Write '1' to clear HSYNC_IRQ.
17 VSYNC_IRQ	Interrupt status bit. Write '1' to clear VSYNC_IRQ.
16 FIELD_IRQ	Interrupt status bit. Write '1' to clear FIELD_IRQ.
15 RESERVED	This field is reserved.
14 ECC_EN	When this bit is set ECC errors generate ERROR_IRQ and the nature of ECC error gets reflected on the ERROR_CODE bit field.
13 ERROR_EN	Interrupt enable bit for ERROR_IRQ.
12 DMA_END_EN	Interrupt enable bit for DMA_END_IRQ.
11 VSTART_EN	Interrupt enable bit for VSTART_IRQ.
10 HSYNC_EN	Interrupt enable bit for HSYNC_IRQ.
9 VSYNC_EN	Interrupt enable bit for VSYNC_IRQ.
8 FIELD_EN	Interrupt enable bit for FIELD_IRQ.
7–4 ERROR_CODE	Error code. Signals errors that triggered error IRQ. Other values not given are reserved. 0000 No error 0001 DMA arm command given during vertical active, DMA_ACT does not accept the value on IPS bus. 0010 DMA arm command given during vertical blanking when DMA_ACT is set. 0100 Line too long 0101 Too many lines in a field/frame 0110 Line too short 0111 Not enough lines in a field/frame 1000 FIFO overflow 1001 FIFO underflow 1010 One bit ECC error 1011 Two or more bits ECC error
3–1 FORMAT_CTRL	Output pixel data format control. See below or refer to for detailed definition. Here 32-bit means MODE32BIT is set and 16-bit means MODE32BIT is cleared. RGB mode means YUV2RGB_EN is set and YUV mode means YUV2RGB_EN is cleared. C means U or V. Dummy means "don't care" data. 16-bit RGB mode: 3'b000: {R[7:3], G[7:2], B[7:3]};

Table continues on the next page...

VIU3_SCR field descriptions (continued)

Field	Description
	3'b001: {G[4:2], B[7:2], R[7:3], G[7:5]}; 32-bit RGB mode: 3'b000: {alpha, R, G, B}; 3'b001: {alpha, B, G, R}; 3'b010: {R, G, B, alpha}; 3'b011: {B, G, R, alpha}; 16-bit YUV mode: 3'b000: {C,Y}; 3'b001: {Y,C}; 32-bit YUV mode: 3'b000: {dummy, Y, U, V}; 3'b001: {dummy, Y, V, U}; 3'b010: {dummy, U, V, Y}; 3'b011: {dummy, V, U, Y}; 3'b100: {Y, U, V, dummy}; 3'b101: {Y, V, U, dummy}; 3'b110: {U, V, Y, dummy}; 3'b111: {V, U, Y, dummy};
0 SOFT_RESET	Writing 1 to this bit generates an internal reset to all components except registers in the VIU3 block. This bit should be set by software when an error interrupt is detected, and it needs to be cleared by software to release the software reset.

57.4.2 Luminance Coefficients For Red, Green And Blue Matrix (VIU3_LUMA_COMP)

The RGB pixel value is computed using following formulae:

$$\text{Red} = \frac{(Y-16)\cdot \text{Y-RED}}{512} + \frac{(Cr-128)\ \text{CR\_RED}}{512} + \frac{(Cb-128)\ \text{CB\_RED}}{512}$$

$$\text{Green} = \frac{(Y-16)\cdot \text{Y\_GREEN}}{512} + \frac{(Cr-128)\ \text{CR\_GREEN}}{512} + \frac{(Cb-128)\ \text{CB\_GREEN}}{512}$$

$$\text{Blue} = \frac{(Y-16)\cdot \text{Y\_BLUE}}{512} + \frac{(Cr-128)\ \text{CR\_BLUE}}{512} + \frac{(Cb-128)\ \text{CB\_BLUE}}{512}$$

The multiplications with Y_red, Y_green, and Y_blue are unsigned multiplications. The multiplications with Cr_red, Cb_red, Cr_green, Cb_green, Cr_blue, and Cb_blue are signed multiplications. The addition is saturated to prevent overflow.

Address: 400C_9000h base + 4h offset = 400C_9004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Y_RED										Reserved	Y_GREEN				
Reset	1	0	0	1	0	1	0	1	0	0	0	1	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Y_GREEN					Reserved	Y_BLUE									
Reset	1	0	1	0	0	0	1	0	0	1	0	1	0	1	0	0

VIU3_LUMA_COMP field descriptions

Field	Description
31–22 Y_RED	(Y_RED[9:0]) Luminance coefficient for red matrix.
21 RESERVED	This field is reserved.
20–11 Y_GREEN	(Y_GREEN[9:0]) Luminance coefficient for green matrix.
10 RESERVED	This field is reserved.
9–0 Y_BLUE	(Y_BLUE[9:0]) Luminance coefficient for blue matrix.

57.4.3 Chroma Coefficients For Red Matrix (VIU3_CHROMA_RED)

Address: 400C_9000h base + 8h offset = 400C_9008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved					CR_RED											Reserved				CB_RED											
Reset	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VIU3_CHROMA_RED field descriptions

Field	Description
31–27 RESERVED	This field is reserved.
26–16 CR_RED	(CR_RED[10:0]) Cr coefficient for red matrix.
15–12 RESERVED	This field is reserved.
11–0 CB_RED	(CB_RED[11:0]) Cb coefficient for red matrix.

57.4.4 Chroma Coefficients For Green Matrix (VIU3_CHROMA_GREEN)

Address: 400C_9000h base + Ch offset = 400C_900Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					CR_GREEN											Reserved				CB_GREEN											
Reset	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	0	0	0

VIU3_CHROMA_GREEN field descriptions

Field	Description
31–27 RESERVED	This field is reserved.
26–16 CR_GREEN	(CR_GREEN[10:0]) Cr coefficient for green matrix.
15–12 RESERVED	This field is reserved.
11–0 CB_GREEN	(CB_GREEN[11:0]) Cb coefficient for green matrix.

57.4.5 Chroma Coefficients For Blue Matrix (VIU3_CHROMA_BLUE)

Address: 400C_9000h base + 10h offset = 400C_9010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					CR_BLUE											Reserved				CB_BLUE											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

VIU3_CHROMA_BLUE field descriptions

Field	Description
31–27 RESERVED	This field is reserved.
26–16 CR_BLUE	(CR_BLUE[10:0]) Cr coefficient for blue matrix.
15–12 RESERVED	This field is reserved.
11–0 CB_BLUE	(CB_BLUE[11:0]) Cb coefficient for blue matrix.

57.4.6 Base Address Of Every Field/Frame Of Picture In Memory (VIU3_DMA_ADDR)

Address: 400C_9000h base + 14h offset = 400C_9014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VIU3_DMA_ADDR field descriptions

Field	Description
31–3 ADDR	(ADDR[31:3]) Base address of every field of picture in memory used by DMA. Rewrite only after receiving DMA done interrupt and before arming DMA. The lowest 3 bits (bits 26:28 as shown here) of ADDR cannot be set. It is always 3'b0.
2–0 RESERVED	This field is reserved.

57.4.7 Horizontal DMA Increment (VIU3_DMA_INC)

Address: 400C_9000h base + 18h offset = 400C_9018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																INC													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VIU3_DMA_INC field descriptions

Field	Description
31–16 RESERVED	This field is reserved.
15–3 INC	(INC[15:3]) Value of this field should be zero or memory size that one active line occupies in memory. It will be added to the memory mapped rounded address at the end of every line. Memory size of one active line depends on line pixel number. It is

Table continues on the next page...

VIU3_DMA_INC field descriptions (continued)

Field	Description
	PIXEL_COUNT[15:2] + PIXEL_COUNT[1:0] when MODE32BIT=0; PIXEL_COUNT[15:1] + PIXEL_COUNT[0] when MODE32BIT=1; It shall only be configured when DMA is inactive, during vertical blanking.
2–0 RESERVED	This field is reserved.

57.4.8 Input Video Pixel and Line Count (VIU3_INVSZ)

Address: 400C_9000h base + 1Ch offset = 400C_901Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LINEC																PIXELC															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0

VIU3_INVSZ field descriptions

Field	Description
31–16 LINEC	(LINEC[15:0]) Expected number of active lines in each input video field/frame. It shall only be configured when DMA is non-active, during vertical blanking If more lines are found during data receive part, a "too many lines" error interrupt is generated when ERROR_IRQ is set. Redundant lines are discarded. If fewer lines are found during data receive part, a "not enough lines error interrupt is generated when ERROR_IRQ is set.
15–0 PIXELC	(PIXELC[15:0]) Expected number of active pixels in each input video line, it shall be an integer multiple of 4. It shall only be configured when DMA is non-active, during vertical blanking. If more pixels are found during data receive part, a "line too long" error interrupt is generated when ERROR_IRQ isset. Redundant pixels are discarded. If less pixels are found during data receive part, a line too short error interrupt is generated when ERROR_IRQ is set.

57.4.9 High IPM Request Priority Alarm (VIU3_HPRALRM)

Address: 400C_9000h base + 20h offset = 400C_9020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								ALARM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0

VIU3_HPRALRM field descriptions

Field	Description
31–8 RESERVED	This field is reserved.

Table continues on the next page...

VIU3_HPRALRM field descriptions (continued)

Field	Description
7–0 ALARM	(ALARM[15:0]) High priority alarm threshold. When FIFO_FILL (FIFO_FILL means the amount of data that is allowed to accumulate in the DMA FIFO due to the DMA being unable to get access to the bus, i.e. a high watermark) is higher than this value, high priority bus request will be asserted. The high watermark threshold is set in terms of 64-bit words.

57.4.10 Programable Alpha Value (VIU3_ALPHA)

Address: 400C_9000h base + 24h offset = 400C_9024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								ALPHA							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

VIU3_ALPHA field descriptions

Field	Description
31–8 RESERVED	This field is reserved.
7–0 ALPHA	(ALPHA[7:0]) Alpha value used for picture blending. This register is configured during vertical blanking and used from the next video field.

57.4.11 Down Scaling Factor In Horizontal Direction (VIU3_HFACTOR)

Address: 400C_9000h base + 28h offset = 400C_9028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																					FACTOR										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

VIU3_HFACTOR field descriptions

Field	Description
31–11 RESERVED	This field is reserved.
10–0 FACTOR	Down scaling factor at horizontal direction. FACTOR[10:8] (register bits 21:23 as shown here) is used as integer part of the factor. FACTOR[7:0] (register bits 24:31 as shown here) is used as fractional part of the factor.

57.4.12 Down Scaling Factor In Vertical Direction (VIU3_VFACTOR)

Address: 400C_9000h base + 2Ch offset = 400C_902Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																					FACTOR										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

VIU3_VFACTOR field descriptions

Field	Description
31–11 RESERVED	This field is reserved.
10–0 FACTOR	Down scaling factor at vertical direction. FACTOR[10:8] (register bits 21:23 as shown here) is used as integer part of the factor. FACTOR[7:0] (register bits 24:31 as shown here) is used as fractional part of the factor.

57.4.13 Down Scaling Destination Pixel and Line Count (VIU3_VID_SIZE)

Address: 400C_9000h base + 30h offset = 400C_9030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LINEC																PIXELC															
Reset	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0

VIU3_VID_SIZE field descriptions

Field	Description
31–16 LINEC	(LINEC[15:0]) Expected number of lines in each output video frame after down scaling.
15–0 PIXELC	(PIXELC[15:0]) Expected number of pixels in each output video line after down scaling. It shall be a multiple of 2 in 32-bit output mode, and a multiple of 4 in 16-bit output mode.

57.4.14 B/C Adjust Look-up-table Current Address (VIU3_LUT_ADDR)

Address: 400C_9000h base + 34h offset = 400C_9034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						ADDR								Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VIU3_LUT_ADDR field descriptions

Field	Description
31–10 RESERVED	This field is reserved.
9–2 ADDR	(ADDR[9:2]) Current address pointer of the B/C adjust look-up-table. Value of this register increments (by 4) automatically at the end of each LUT_DATA write/read operation. This function allows fast update to the whole look-up-table, to the table of one color component, or even to any random field of the table. Note: ADDR reflects correct address only when clock of B/C adjust block is valid.
1–0 RESERVED	This field is reserved.

57.4.15 B/C Adjust Look-up-table Data Entry (VIU3_LUT_DATA)

Address: 400C_9000h base + 38h offset = 400C_9038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VIU3_LUT_DATA field descriptions

Field	Description
31–0 DATA	(DATA[31:0]) B/C adjust look-up-table data entry. Data in this register is actually written/read to/from the address pointed to by the current LUT_ADDR value in the table. Note: DATA reflects correct data value only when clock of B/C adjust block is valid.

57.4.16 Extended Configuration Register (VIU3_EXT_CONFIG)

Address: 400C_9000h base + 3Ch offset = 400C_903Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			CS_EN	LENDIAN	0	RGB2YUV_EN	DE_VALID	INP_FORMAT			PCLK_POL	VSYNC_POL	HSYNC_POL	DE_POL	HMIRROR_EN
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

VIU3_EXT_CONFIG field descriptions

Field	Description
31–13 RESERVED	This field is reserved.
12 CS_EN	Chroma swap enable bit. It's used to control how combining Y and C when output format is YUV422 mode. By default, for YUV422 mode(for example, assume 4 pixels per line here), the data in memory is Y0U0, Y1V0, Y2U2, Y3V2. When CS_EN is set, it becomes Y0V0, Y1U0, Y2V2, Y3U2. An application senario of CS_EN is that when mirror is on, the data will be Y3V2, Y2U2, Y1V0, Y0U0 by default. The first chrmo is V, instead of U. If CS_EN is set, the output data become Y3U2, Y2V2, Y1U0, Y0V0. 0 Chroma swap is disabled. 1 Chroma swap is enabled.
11 LENDIAN	Data endian control bit. This bit controls the endian of the IPM data bus. 0 Big endian 1 Little endian
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 RGB2YUV_EN	RGB to YUV conversion enable
8 DE_VALID	External data enable indicator is valid at parallel input modes. 0 DE is invalid 1 DE is valid
7–5 INP_FORMAT	Input video format select bits. It should be set correctly according to the video input. 000 10/8bit ITU stream 001 24bit parallel YUV. Normally it is YUV444 010 Reserved

Table continues on the next page...

VIU3_EXT_CONFIG field descriptions (continued)

Field	Description
	011 Reserved 100 24bit parallel RGB. It is RGB888 101 8bit serial RGB. It is RGB888 110 18bit parallel RGB. It is RGB666 111 16bit parallel RGB. It is RGB565
4 PCLK_POL	Pixel clock polarity control bit. pix_clk will be reversed when the bit is set. 0 Active high 1 Active low
3 VSYNC_POL	Vsync polarity control bit. Vsync will be reversed when the bit is set. 0 Active high 1 Active low
2 HSYNC_POL	Hsync polarity control bit. Hsync will be reversed when the bit is set. 0 Active high 1 Active low
1 DE_POL	Data enable polarity control bit. DE will be reversed when the bit is set. 0 Active high 1 Active low
0 HMIRROR_EN	Horizontal mirror enable.

57.4.17 Red, Green and Blue Coefficients for Luminance component (VIU3_RGB_Y)

The YUV pixel value is computed using the following formulae:

$$Y = \frac{RY \cdot R}{512} + \frac{GY \cdot G}{512} + \frac{BY \cdot B}{512} + 16$$

$$U = \left(-\frac{RU \cdot R}{512}\right) + \left(-\frac{GU \cdot G}{512}\right) + \frac{BU \cdot B}{512} + 128$$

$$V = \frac{RV \cdot R}{512} - \frac{GV \cdot G}{512} - \frac{BV \cdot B}{512} + 128$$

All cofficients are unsigned.

Address: 400C_9000h base + 40h offset = 400C_9040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		R_Y								Reserved		G_Y			
Reset	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	G_Y					Reserved					B_Y					
Reset	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0

VIU3_RGB_Y field descriptions

Field	Description
31–30 RESERVED	This field is reserved.
29–22 R_Y	(R_Y[7:0]) Red coefficient for luminance component
21–20 RESERVED	This field is reserved.
19–11 G_Y	(G_Y[8:0]) Green coefficient for luminance component
10–6 RESERVED	This field is reserved.
5–0 B_Y	(B_Y[5:0]) Blue coefficient for luminance component

57.4.18 Red, Green and Blue Coefficients for Chroma U component (VIU3_RGB_U)

Address: 400C_9000h base + 44h offset = 400C_9044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved			R_U							Reserved			G_U								Reserved			B_U							
Reset	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	1	1	1	0	0	0	0	1

VIU3_RGB_U field descriptions

Field	Description
31–29 RESERVED	This field is reserved.
28–22 R_U	(R_U[6:0]) Red coefficients for chroma U component
21–19 RESERVED	This field is reserved.
18–11 G_U	(G_U[7:0]) Green coefficients for chroma U component
10–8 RESERVED	This field is reserved.
7–0 B_U	(B_U[7:0]) Blue coefficients for chroma U component

57.4.19 Red, Green and Blue Coefficients for Chroma V component (VIU3_RGB_V)

Address: 400C_9000h base + 48h offset = 400C_9048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		R_V								Reserved			G_V		
Reset	0	0	1	1	1	0	0	0	0	1	0	0	0	1	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	G_V					Reserved					B_V					
Reset	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0

VIU3_RGB_V field descriptions

Field	Description
31–30 RESERVED	This field is reserved.
29–22 R_V	(R_V[7:0]) Red coefficients for chroma V component
21–19 RESERVED	This field is reserved.
18–11 G_V	(G_V[7:0]) Green coefficients for chroma V component
10–6 RESERVED	This field is reserved.
5–0 B_V	(B_V[5:0]) Blue coefficients for chroma V component

57.5 Functional Description

The VIU3 accepts parallel RGB, serial RGB, parallel YUV and ITU-R BT.656 compatible video streams on its parallel interface, decodes it and optionally performs processes like scaling, brightness and contrast adjust, RGBYUV to YUVRGB conversion, and de-interlace (weaving), then stores the result video stream to the system memory which can then be displayed by a display controller, or post processed.

Functions of the VIU3 are designed in a way that they can be flexibly enabled or disabled by software. But there are a few limitations as listed below:

- The down scaler works on YUV 4:2:2 format or RGB888 formats, so to enable the scalar, decoder shall be configured in YUV 4:2:2 or RGB888 mode.
- To use the scaler, progressive video input shall be used for display quality's sake, and the de-interlace shall be disabled.

57.5.1 Input Formats

VIU3 accepts parallel RGB, serial RGB, parallel YUV and ITU-R BT.656 compatible video streams on its parallel interface. Below is the description for the timing of input video formats.

57.5.1.1 ITU656

The ITU-R BT.656-4 recommendation describes the means of interconnecting digital television equipment operating on the 525-line or 625-line standards and combines with the 4:2:2 encoding parameters as defined in the ITU-R BT.601 recommendation.

The data stream structure on ITU-R BT.656-4 interface is shown in the following figure. There are two timing reference signals, one at the beginning of each video data block (start of active video, SAV) and one at the end of each video data block (end of active video, EAV).

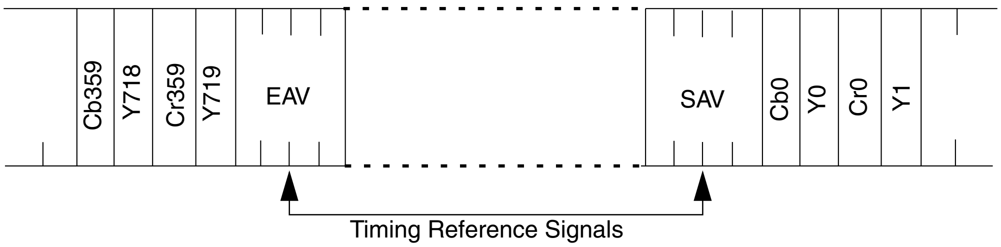


Figure 57-21. Interface Data Stream of ITU-R BT.656-4

Each timing reference signal consists of a four-word sequence in the following format: FF 00 00 XY. Values are expressed in hexadecimal notation. Value FF and 00 are reserved to be used in the timing reference signals.

The first three words are a fixed preamble. The fourth word contains information defining field 2 identification, the state of field blanking, and the state of line blanking. The assignment of bits within the timing reference signal is shown in the table below.

Table 57-22. Video Timing Reference Codes

Data Bit Number	First Word (FF)	Second Word (00)	Third Word (00)	Fourth Word (XY)
9(MSB)	1	0	0	1
8	1	0	0	F(0: field 1, 1: field 2)
7	1	0	0	V(0: elsewhere, 1: field blanking)
6	1	0	0	H(0: in SAV, 1: in EAV)

Table continues on the next page...

Table 57-22. Video Timing Reference Codes (continued)

Data Bit Number	First Word (FF)	Second Word (00)	Third Word (00)	Fourth Word (XY)
5	1	0	0	P3
4	1	0	0	P2
3	1	0	0	P1
2	1	0	0	P0
1	1	0	0	0
0	1	0	0	0

In above table, bits P0, P1, P2, P3 have states dependent on the states of the bits F, V and H. At the receiver side this arrangement permits one-bit errors to be corrected and two-bit errors to be detected.

Progressive video is also composed of fields; but instead of containing even lines in one field and odd lines in the other, the fields contain contiguous lines, i.e., field0 contains lines 1,2,3,... and field1 contains rest of the lines starting from (last line number of field0 + 1). And the F bit cannot be ignored in case of progressive frame.

Refer to the ITU-R BT.656-4 recommendation for more details.

57.5.1.2 Parallel Input Format

VIU3 also accepts parallel input video formats. It includes 24bit RGB888, YUV444, 18bit RGB666 and 16bit RGB565 formats. Their data streams are similar. The following caveats apply to the timing diagram below:

- FID must change state 3 Hsync pulses into Vsync.
- The first -//- break represents at least 10 lines.
- A field should contain more than 16 active lines.
- Hsync pulse should be longer than 60 pixel clock cycles.
- A active line should contain more than 24 pixels.

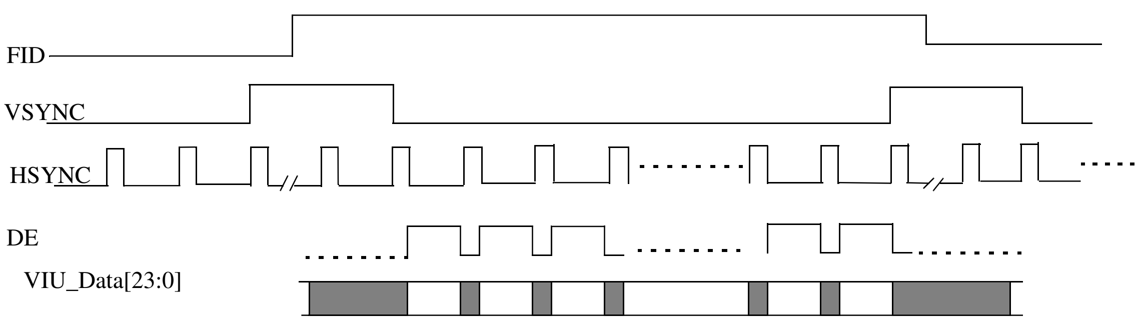


Figure 57-22. Parallel Input Timing

57.5.1.3 Parallel YC Format

Parallel YC video should be in YUV222 format. The timing diagram is provided below.

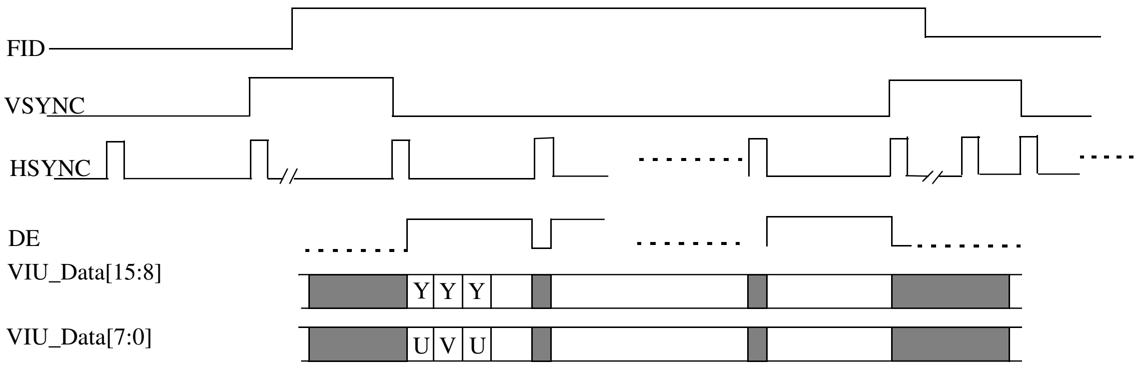


Figure 57-23. Parallel YC Timing

57.5.1.4 Serial RGB888 Format

The timing diagram of serial RGB888 is shown below.

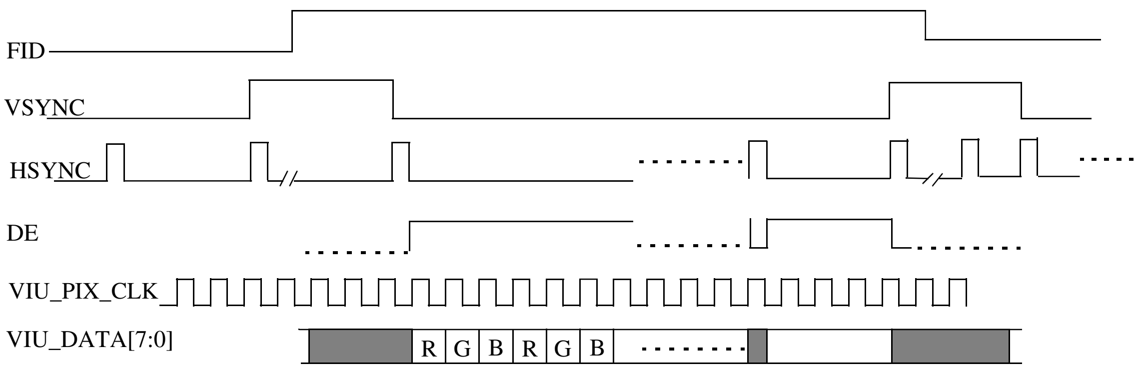


Figure 57-24. Serial RGB888 Timing

57.5.2 Input Synchronizer

The Synchronizer block (1) captures ITU656 protocol signals from the interface, and synchronizes them to the IPG_CLK domain.

57.5.3 Decoder

The ITU Decoder block (2) detects the ITU656 timing reference signal (consists of a four-word sequence in the following format: FF-00-00-XY) and extracts HSYNC, VSYNC, field number signals and video data from the ITU data stream. The format of active pixel data from the ITU stream is YUV 4:2:2. The decoder block can directly send out this YUV 4:2:2 data, or interpolate it to YUV 4:4:4 format and send it out. This is determined by the MODE444 field in the SCR register. This bit shall be set if the down-scaler is enabled, because the down-scaler works on YUV 4:4:4 data format.

57.5.4 Scaling

VIU3 supports up to 1/8 horizontal and vertical down-scaling, as well as 2x horizontal up-scaling.

57.5.4.1 Down Scaling

The Down-scaler block (3) performs down-scaling to the incoming video stream. It is able to scale down the input video stream by a fractional scaling ratio up to 1/8. The down-scaler block can be enabled/disabled by software via the SCALER_EN bit in the SCR register. To use the down-scaler video data extracted from ITU decoder shall be in YUV 4:4:4 format.

The down-scaler uses a bi-linear filter with simple linear interpolation between the two nearest neighbors, on both horizontal and vertical direction. Scaling is firstly done on the horizontal direction (called XScale), and then the vertical direction (called YScale). Different scaling factors are supported on horizontal direction and vertical direction.

Assuming the scaling factors are (factor_h, factor_v), in theory for every pixel (x, y) of the scaled picture the coordinates of the corresponding pixel in the incoming picture (x1, y1) can be calculated by multiplying the coordinates with the scaling factors, say (x1 = x * factor_h, y1 = y * factor_v). Now because the scaling factors can be fractions, the coordinates (x1, y1) are not integer anymore, they are fractional values. The scalers use the integer part of (x1, y1) to find the two neighboring pixels from the incoming picture as input to the filters, and the fractional part to derive the weighting factors for interpolation.

The scaling factors, (factor_h, factor_v), are both 11 bit with the lower 8 bit as the factional part and the highest 3 bit the integer part. It is capable of scaling the input picture by up to 8, in steps of 1/256. It is by 8 if the factor is programmed as all zeros.

Instead of using multipliers to calculate (x1, y1), two 20-bit phase accumulator is used to step through the source pixels/or lines, where the lower 8 bit is the fractional part. For every output pixel/or line the phase adder is added to the accumulator. With the 12 bit integer part of the accumulator, up to 4096 source pixels/or lines can be supported.

So for each target pixel, the current accumulator position is used to determine how the pixel is going to be produced. As an example, accumulator value 0x123.3a means the step position is between pixel[0x123] and pixel[0x124], and the output pixel will be calculated by (pixel[0x124] * (0x100 - 0x3a) + pixel[0x123] * 0x3a) >> 8. The same concept is used for horizontal and vertical scaling.

Because of the vertical scaling, a line buffer is used for each color component to store the data from horizontal scaling, it stores one line of data. Size of the line buffer determines the maximum video output size after scaling.

The scaling factors are programmed via the HFACTOR and VFACTOR registers. Video size after scaling is programmed via the VID_SIZE register.

Three scalers are instantiated in the down-scaler block, one for each component.

57.5.4.2 Up-scaling

The theory of up-scaling is the same with down-scaling. When HFACTOR is smaller than 1, up-scaling is implemented by VIU3 scaler. But note that VIU3 does not support vertical up-scaling, and HFACTOR should not be less than 0.5.

Note

The scaled pixel count per line shall be an integer multiple of 2 in 32-bit output mode, or 4 in 16-bit output mode. The user shall divide the input pixel count by the horizontal scaling factor and truncate the result to a multiple of 2 or 4.

Note

To achieve good display quality, progressive video input shall be used if down-scaling is needed.

57.5.5 Brightness and Contrast Adjust

The B/C Adjust block (4) performs brightness and contrast adjustment on the input video stream via three internal look-up-tables, one table per color component. Each table contains 256 8-bit entries, it maps every incoming pixel to the value of one of its entries

according to the original value of the pixel. This feature allows the user to adjust brightness and/or contrast of the incoming picture according to three arbitrary adjustment curves, one adjustment curve per color component.

To use this feature, the B/C adjust look-up-table shall be programmed in software in the following format.

Table 57-23. B/C Adjust Look-up-table Format

Color Component	BC LUT Offset	Local Address [1:0]			
		00	01	10	11
Y	0x000	$BC0_Y$	$BC1_Y$	$BC2_Y$	$BC3_Y$
	......				
	0x0FC	$BC252_Y$	$BC253_Y$	$BC254_Y$	$BC255_Y$
U	0x100	$BC0_U$	$BC1_U$	$BC2_U$	$BC3_U$
	......				
	0x1FC	$BC252_U$	$BC253_U$	$BC254_U$	$BC255_U$
V	0x200	$BC0_V$	$BC1_V$	$BC2_V$	$BC3_V$
	......				
	0x2FC	$BC252_V$	$BC253_V$	$BC254_V$	$BC255_V$

Two registers are provided to program the look-up-table, they are LUT_ADDR and LUT_DATA. LUT_ADDR is the current address pointer, which always points to the current table offset to be programmed. It can be set by software, and it increases (by 4) automatically when each word is written to the table, and it falls back to 0x000 if the current value is 0x2FC and the last word is written to the table. LUT_DATA is the table data entry, data written to this register will be stored into the table, to the address pointed to by LUT_ADDR. This register shall only be written by 4-byte word. With the combination of these two registers the user can program the B/C look-up-table conveniently, either program the whole table, or reprogram the table for one color component only, or even change one single arbitrary word in the table.

The B/C adjust can be enabled/disabled by software via the BC_EN bit in the SCR register.

57.5.6 YUV to RGB Conversion

The YUV2RGB block (5) is used to convert YUV (4:2:2 or 4:4:4) to RGB (888 or 565). The coefficients of the YUV to RGB conversion matrix are programmed via four registers. When the input is YUV 4:2:2, it is interpolated to YUV 4:4:4 first.

57.5.7 Round and Dither

In RGB565 output mode, when pixel data is converted from RGB888 to RGB565 the image is anamorphic more or less, because of losing color information conveyed by the least significant two or three bits of the original color components, which are dropped. VIU3 block provides two simple algorithms, round and dither, to compensate this color information loss.

57.5.7.1 Round

In round mode, VIU3 will round to 1 in LSB if the decimal fraction is bigger than 0.5 and ignore the smaller fraction when ROUND_ON is set in the SCR register.

57.5.7.2 Dither

Dither is a little more complex but better than round for recovering image. It is a statistical compensation algorithm. It does not render all pixels with the same grey or color level, but some with the lower one, and some with a color level of 1 LSB more. The selection of adding one LSB or not depends on the position of the pixel on the screen.

The figure below shows the implementation of dither in the VIU3 block.

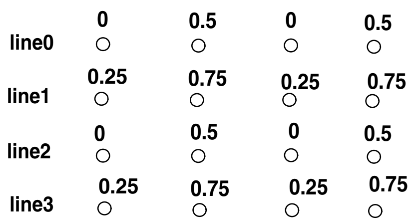


Figure 57-25. Dither Implementation

The number above the pixel position in the diagram is the compensation value for this pixel. When pixels have a value of 0.25, they are rendered 0 in 75% of the pixels and 1 in 25% of pixels. This averages to 0.25. Similarly, pixel values of 0.5, 0.75, and 1.0 are rendered 50%, 75% and 100% of the pixels as 1. For human eyes, this rendering result of dither makes the holistic image smoother and closer to the original one.

57.5.8 Output Formatter

The Output Formatter block (6) accepts YUV or RGB data from the YUV2RGB block, arranges them in expected format, and then sends it to the DMA engine. The FORMAT_CTRL field in SCR register controls how the VIU3 stores data to system memory; see the following figure for details.>

Table 57-24. VIU3 Output Data Stream Format

Pixel Format	FORMAT _CTRL[1]	Local Address [2:0]							
		000	001	010	011	100	101	110	111
RGB 565	000	R0[7:3], G0[7:2], B0[7:3]		R1[7:3], G1[7:2], B1[7:3]		R2[7:3], G2[7:2], B2[7:3]		R3[7:3], G3[7:2], B3[7:3]	
	001[2]	G0[4:2], B0[7:3], R0[7:3], G0[7:5]		G1[4:2], B1[7:3], R1[7:3], G1[7:5]		G2[4:2], B2[7:3], R2[7:3], G2[7:5]		G3[4:2], B3[7:3], R3[7:3], G3[7:5]	
RGB 8888	000	A0	R0	G0	B0	A1	R1	G1	B1
	001	A0	B0	G0	R0	A1	B1	G1	R1
	010	R0	G0	B0	A0	R1	G1	B1	A1
	011	B0	G0	R0	A0	B1	G1	R1	A1
YUV 4:2:2	000	U0	Y0	V0	Y1	U2	Y2	V2	Y3
	001	Y0	U0	Y1	V0	Y2	U2	Y3	V2
YUV 4:4:4	000	dummy	Y0	U0	V0	dummy	Y1	U1	V1
	001	dummy	Y0	V0	U0	dummy	Y1	V1	U1
	010	dummy	U0	V0	Y0	dummy	U1	V1	Y1
	011	dummy	V0	U0	Y0	dummy	V1	U1	Y1
	100	Y0	U0	V0	dummy	Y1	U1	V1	dummy
	101	Y0	V0	U0	dummy	Y1	V1	U1	dummy
	110	U0	V0	Y0	dummy	U1	V1	Y1	dummy
	111	V0	U0	Y0	dummy	V1	U1	Y1	dummy

1. All values not shown in the table shall be considered as reserved and shall not be used.
2. This format is basically intended for data communication with any little-endian peripheral in the system, assuming big-endian system memory is used.

NOTE

RGB666 input can be stored in memory in RGB565 or RGB8888 formats. When stored as RGB565, the R[0] and B[0] component of RGB666 input will be discarded. When stored as RGB8888, the RGB666 input data will be reflected in memory as {R[5:0], 2'b00, G[5:0], 2'b00, B[5:0], 2'b00}. The position of Alpha bit will depend on the FORMAT_CTRL setting.

57.5.9 High Priority Alarm

FIFO_FILL is a status field which indicates the amount of data in the FIFO. The HI_PRIO_ALARM register is a threshold such that, when the data number in the FIFO is larger than HI_PRIO_ALARM, the DMA request will be high-priority on the AMBA crossbar side. The purpose of this register is to enhance DMA performance. Normally the reset value should be sufficient; however, if the FIFO is seen to overflow, the user should lower the threshold in order to allow the DMA to service the FIFO more frequently.

57.5.10 DMA and De-interlace

The DMA engine block (7) functions as the FIFO controller and DMA engine. It stores the data from the output formatter block into a FIFO and then writes them to system memory via the IPM interface.

VIU3 block has an embedded DMA. When video data is converted to RGB format and placed into a 256 × 64 bit FIFO, it waits to be transferred to memory by the internal DMA.

After doing some necessary register configuration, such as coefficients, INVSZ and DMA_ADDR, user can activate DMA by setting the DMA_ACT of the status and configuration register. But note that DMA_ACT can be configured only during vertical blanking since the VIU3 block will not transfer a fragment of video field to memory. If it is configured during field active time, DMA transfer cannot be started and an error interrupt will be asserted.

The VIU3 block asserts a transfer request when there is enough data in the FIFO for one transfer. Normally one transfer conveys 32 bytes from FIFO to memory. At the end of a line, all remaining data is transferred, though it may not be as much as 32 bytes.

VIU3 also provides a simple way to de-interlace for interlaced or pseudo-interlaced video images. It is implemented by setting the DMA_ADDR and DMA_INC registers.

The figure below shows the implementation of de-interlace. The value of DMA_INC is added to the rounded address at the end of every active line. So, when DMA_INC is zero, pixel data is stored in memory line by line, meaning de-interlace is off, as shown in figure (1) and (2); otherwise, when DMA_INC equals to one-line memory mapped pixel size and DMA_ADDR(2) = DMA_ADDR(1) + DMA_INC, the odd field and even field will be merged into one frame in memory, as shown in figure (3).

Here DMA_ADDR(1) means base address of field 1, and DMA_ADDR(2) means base address of field 2. Memory mapped line size means memory size that is occupied by one active line pixels. It depends on pixel number of one line and video data format (RGB888 or RGB565). See the register description section for more details.

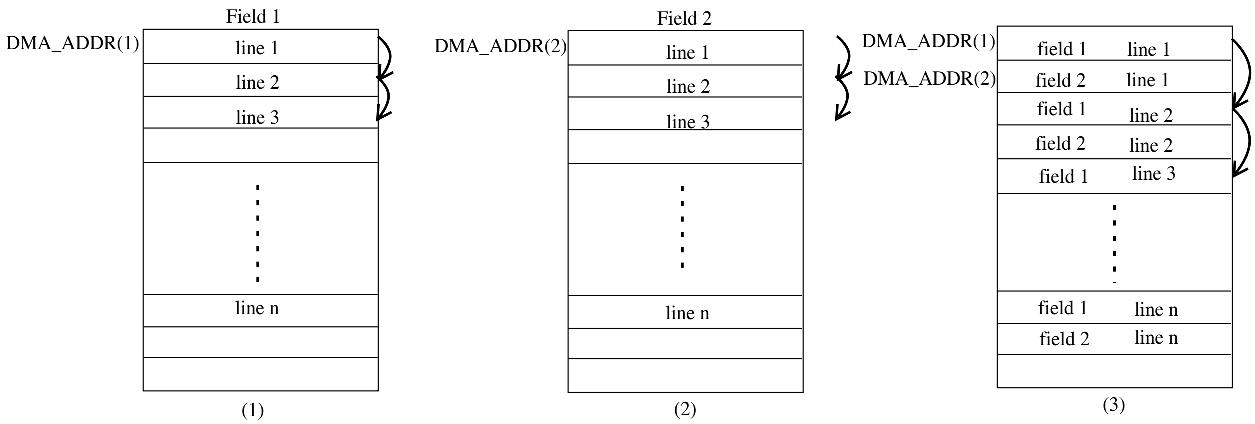


Figure 57-26. Implementation of De-interlace

When the HMIRROR_EN bit is set in the EXT_CONFIG register, the image can be mirrored in the horizontal direction, as illustrated below.

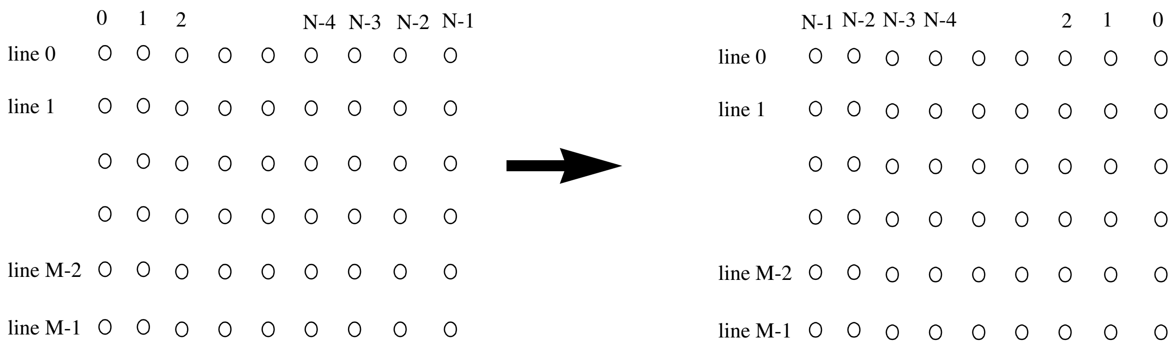


Figure 57-27. Horizontal Image Mirror

57.5.11 Error Case

Normally, the user should provide a standard and totally ITU-compatible video stream to the VIU3 block. However, it is difficult to avoid unexpected errors all the time. VIU3 can manage error cases like ECC error, line too long, line too short, too many lines, not enough lines in a field, and FIFO overflow.

- ECC error: ITU stream provides 4-bit error correcting code P[3:0] in its SAV and EAV. It is decoded in VIU3 to use the correct field number, horizontal sync and vertical sync bits. It can correct one bit errors and find two bit errors. When an ECC error is found, an interrupt is asserted.
- Line too long error: When pixels of active line is longer than PIXELC, a line too long error interrupt is asserted and redundant pixels are discarded.

- Too many lines error: When active lines of a field are bigger than LINEC, a too many lines error interrupt is asserted and redundant lines are discarded.
- Line too short error: When pixels of active line is less than PIXELC, a line too short error interrupt is asserted.
- Not enough line error: When active lines of a field is less than LINEC, a not enough line error interrupt is asserted.
- FIFO overflow error: If the system bus is blocked for a long time, video data is stored in FIFO and causes FIFO overflow. When FIFO overflow occurs, an interrupt is asserted and incoming data is discarded until FIFO works normally again. Current field is jumbled. However, VIU3 recovers to work at the next field if the bus is unblocked at that time.
- FIFO underflow: When the FIFO is read when it is empty, a FIFO underflow error interrupt is asserted. Normally, this error should not occur.

VIU3 can manage the above error cases to a certain extent. However, when it does not recover to a working state the user should write the SOFT_RESET bit of the status and configuration register to reset the VIU3 block.

57.6 Initialization/Application Information

Initialization steps and startup information are given below.

57.6.1 Initialization Information

When the VIU3 block comes out of reset, software should implement the following steps to start this block.

1. Program the SCR register to set the VIU3 to the desired operation mode
   - To enable YUV 4:2:2 to 4:4:4 interpolation in the ITU decoder, set the MODE444 bit[2]
   - To enable the down-scaler, set the SCALER_EN bit
   - To enable B/C adjust, set the BC_EN bit

2. MODE444 bit shall not be set when the down-scaler is enabled.

- To enable RGB565 output mode, set the YUV2RGB_EN bit and clear the MODE32BIT bit. Optionally set the DITHER_ON bit or the ROUND_ON bit to enable dither or round[3]
- To enable RGB8888 output mode, set the YUV2RGB_EN and MODE32BIT bits
- To enable YUV output mode, clear the YUV2RGB_EN bit

2. Set the FORMAT_CTRL field so that the VIU3 outputs data in the correct format. Configure the input video size via the INVSZ register.
3. If it is desired to use RGB output, configure YUV to RGB conversion coefficients or use the default values after reset.
4. If it is desired to use the down scaling function, program the down scaling factors and destination video size after scaling.
5. If it is desired to use the B/C adjust function, program the B/C adjust look-up-table via the LUT_DATA and LUT_ADDR registers.
6. Configure the HPRALRM and ALPHA registers if necessary.
7. Set the VSYNC_EN and/or FIELD_EN bits in the SCR register to enable vsync or field interrupt. Meanwhile, disable error interrupt.
8. When software receives a vsync interrupt and/or field interrupt, read FIELD_NO bit of the SCR register[4].
9. According to the FIELD_NO bit, program the DMA_ADDR register. This is the field start address in system memory, or frame start address in progressive video input mode.
10. If it is desired to use the de-interlace (weaving) function, program the line start address offset value in the DMA_INC register[5].
11. Clear error status first if necessary.
12. Write the DMA_ACT bit of the SCR register to start FIFO and DMA transfer. This operation actually starts the VIU3 to operate.

3. If DITHER_ON or ROUND_ON are both set, only round will be enabled.
4. Reading the FIELD_NO bit is optional, especially in progressive video input mode.
5. Progressive video input shall be used when down scaling is enabled for better display quality.

57.6.2 Application Information

Normally, the user shall not change the register values of the function enable bits, input/output video size, color conversion coefficients, DMA_INC, MODE32BIT, scaling factors and B/C adjust look-up-table contents after VIU3 is started up. When the video input source is changed, the user should reset the VIU3 and re-configure the related registers.

57.6.2.1 Register Configuration Timing Window

As mentioned above, dynamic configuration of registers is not recommended because it may cause error if it is not configured in a certain timing window, especially for INVSZ and DMA_INC.

Register configuration timing window is shown in the below diagram. All registers, except for the SOFT_RESET bit, are recommended to be configured during vertical blanking, after DMA transfer is done and the field identification bit is changed (field interrupt is asserted).

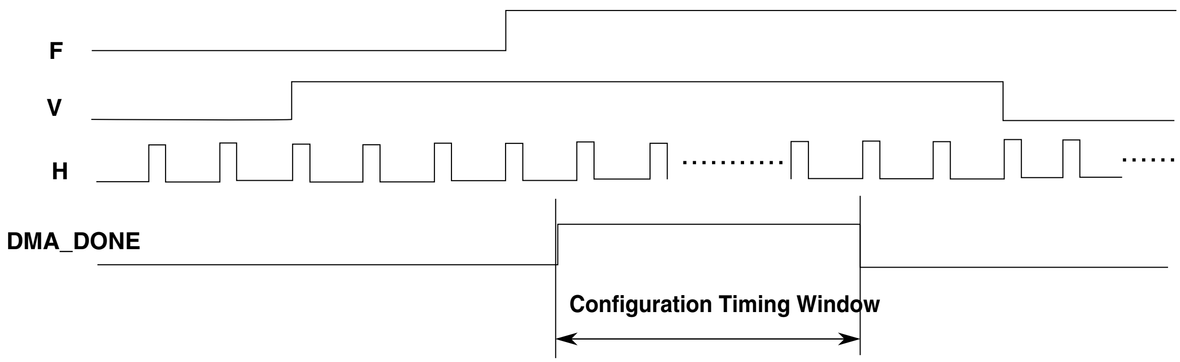


Figure 57-28. Register Configuration Timing Window

Chapter 58
Timing Controller TCON

58.1 Introduction

The Timing Controller module (TCON) provides an alternative interface for the DCU3 that provides RGB data and timing signals for "raw" TFT panels which have no embedded TCON.

58.1.1 Features

- Flexible timing generation unit supporting 12 timing signal channels
- Support bit mapping of 8-bit or 6-bit color depth
- Blanking of RGB data during inactive period (driven to all "0" or all "1")

58.1.2 Modes of Operation

The TCON has 2 operation modes:

- Bypass mode: the input signals are passed through, and the TCON is turned off
- TTL mode: the TCON is functional, driving parallel RGB output, and the TCON timing signals

58.2 External Signal Descriptions

The TCON has the following external signals:

Table 58-1. TCON External Signals

Signal	Description	I/O	Reset
data_out[25:0]	pixel data and clock out	O	0
tcon_out[11:0]	12 tcon timing signals	O	0

58.3 Memory map and register definition

The memory map for the TCON module is given below. The total address for each register is the sum of the base address for the TCON module and the address offset for each register.

TCON memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_D000	TCON control1 register (TCON0_CTRL1)	32	R/W	0000_000Ch	58.3.1/3299
4003_D004	Bit map control register (TCON0_BMC)	32	R/W	0000_0000h	58.3.2/3301
4003_D008	Comparator 0 configure register (TCON0_COMP0)	32	R/W	0000_0FFFh	58.3.3/3302
4003_D00C	Comparator 1 configure register (TCON0_COMP1)	32	R/W	0000_0FFFh	58.3.4/3303
4003_D010	Comparator 2 configure register (TCON0_COMP2)	32	R/W	0000_0FFFh	58.3.5/3304
4003_D014	Comparator 3 configure register (TCON0_COMP3)	32	R/W	0000_0FFFh	58.3.6/3305
4003_D018	Comparator 0 compare value mask register (TCON0_COMP0_MSK)	32	R/W	0000_0FFFh	58.3.7/3305
4003_D01C	Comparator 1 compare value mask register (TCON0_COMP1_MSK)	32	R/W	0000_0FFFh	58.3.8/3306
4003_D020	Comparator 2 compare value mask register (TCON0_COMP2_MSK)	32	R/W	0000_0FFFh	58.3.9/3306
4003_D024	Comparator 3 compare value mask register (TCON0_COMP3_MSK)	32	R/W	0000_0FFFh	58.3.10/ 3307
4003_D028	Pulse 0 configure register (TCON0_PULSE0)	32	R/W	0FFF_0FFFh	58.3.11/ 3307
4003_D02C	Pulse 1 configure register (TCON0_PULSE1)	32	R/W	0FFF_0FFFh	58.3.12/ 3308
4003_D030	Pulse 2 configure register (TCON0_PULSE2)	32	R/W	0FFF_0FFFh	58.3.13/ 3310
4003_D034	Pulse 3 configure register (TCON0_PULSE3)	32	R/W	0FFF_0FFFh	58.3.14/ 3311
4003_D038	Pulse 4 configure register (TCON0_PULSE4)	32	R/W	0FFF_0FFFh	58.3.15/ 3312
4003_D03C	Pulse 5 configure register (TCON0_PULSE5)	32	R/W	0FFF_0FFFh	58.3.16/ 3313

Table continues on the next page...

TCON memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_D040	Pulse 0 compare value mask register (TCON0_PULSE0_MSK)	32	R/W	0FFF_0FFFh	58.3.17/ 3314
4003_D044	Pulse 1 compare value mask register (TCON0_PULSE1_MSK)	32	R/W	0FFF_0FFFh	58.3.18/ 3315
4003_D048	Pulse 2 compare value mask register (TCON0_PULSE2_MSK)	32	R/W	0FFF_0FFFh	58.3.19/ 3315
4003_D04C	Pulse 3 compare value mask register (TCON0_PULSE3_MSK)	32	R/W	0FFF_0FFFh	58.3.20/ 3316
4003_D050	Pulse 4 compare value mask register (TCON0_PULSE4_MSK)	32	R/W	0FFF_0FFFh	58.3.21/ 3316
4003_D054	Pulse 5 compare value mask register (TCON0_PULSE5_MSK)	32	R/W	0FFF_0FFFh	58.3.22/ 3317
4003_D058	Function control register 0 (TCON0_SMX0)	32	R/W	0000_0000h	58.3.23/ 3318
4003_D05C	Function control register 1 (TCON0_SMX1)	32	R/W	0000_0000h	58.3.24/ 3319
4003_D060	Function control registers 2 (TCON0_SMX2)	32	R/W	0000_0000h	58.3.25/ 3321
4003_D064	Function control register 3 (TCON0_SMX3)	32	R/W	0000_0000h	58.3.26/ 3322
4003_D068	Function control register 4 (TCON0_SMX4)	32	R/W	0000_0000h	58.3.27/ 3324
4003_D06C	Function control register 5 (TCON0_SMX5)	32	R/W	0000_0000h	58.3.28/ 3326
4003_D070	Function control register 6 (TCON0_SMX6)	32	R/W	0000_0000h	58.3.29/ 3327
4003_D074	Function control register 7 (TCON0_SMX7)	32	R/W	0000_0000h	58.3.30/ 3329
4003_D078	Function control register 8 (TCON0_SMX8)	32	R/W	0000_0000h	58.3.31/ 3330
4003_D07C	Function control register 9 (TCON0_SMX9)	32	R/W	0000_0000h	58.3.32/ 3332
4003_D080	Function control register 10 (TCON0_SMX10)	32	R/W	0000_0000h	58.3.33/ 3334
4003_D084	Function control register 11 (TCON0_SMX11)	32	R/W	0000_0000h	58.3.34/ 3335
4003_D088	Function control register 12 (TCON0_SMX12)	32	R/W	0000_0000h	58.3.35/ 3337
4003_D08C	Function control register 13 (TCON0_SMX13)	32	R/W	0000_0000h	58.3.36/ 3338
4003_D090	TCON output mux control low (TCON0_OMUX_LOW)	32	R/W	0000_0000h	58.3.37/ 3340
4003_D094	TCON output mux control high (TCON0_OMUX_HIGH)	32	R/W	0000_0000h	58.3.38/ 3341

Table continues on the next page...

TCON memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_D098	TCON look up table 0 (TCON0_LUT0)	32	R/W	FFFF_005Ch	58.3.39/ 3342
4003_D09C	TCON look up table 1 (TCON0_LUT1)	32	R/W	FFFF_005Ch	58.3.40/ 3343
4003_D0A0	TCON look up table 2 (TCON0_LUT2)	32	R/W	FFFF_005Ch	58.3.41/ 3343
4003_D0A4	TCON look up table 3 (TCON0_LUT3)	32	R/W	FFFF_005Ch	58.3.42/ 3343
4003_D0A8	TCON look up table 4 (TCON0_LUT4)	32	R/W	FFFF_005Ch	58.3.43/ 3344
4003_D0AC	TCON look up table 5 (TCON0_LUT5)	32	R/W	FFFF_005Ch	58.3.44/ 3344
4003_D0B0	TCON look up table 6 (TCON0_LUT6)	32	R/W	FFFF_005Ch	58.3.45/ 3344
4003_D0B4	TCON look up table 7 (TCON0_LUT7)	32	R/W	FFFF_005Ch	58.3.46/ 3345
4003_D0B8	TCON look up table 8 (TCON0_LUT8)	32	R/W	FFFF_005Ch	58.3.47/ 3345
4003_D0BC	TCON look up tables 9 (TCON0_LUT9)	32	R/W	FFFF_005Ch	58.3.48/ 3345
4003_D0C0	TCON look up table 10 (TCON0_LUT10)	32	R/W	FFFF_005Ch	58.3.49/ 3346
4003_D0C4	TCON look up table 11 (TCON0_LUT11)	32	R/W	FFFF_005Ch	58.3.50/ 3346
4003_D0C8	TCON look up table 12 (TCON0_LUT12)	32	R/W	FFFF_005Ch	58.3.51/ 3346
4003_D0CC	TCON look up table 13 (TCON0_LUT13)	32	R/W	FFFF_005Ch	58.3.52/ 3347
4003_D104	TCON control2 register (TCON0_CTRL2)	32	R/W	0000_0002h	58.3.53/ 3347
400B_D000	TCON control1 register (TCON1_CTRL1)	32	R/W	0000_000Ch	58.3.1/3299
400B_D004	Bit map control register (TCON1_BMC)	32	R/W	0000_0000h	58.3.2/3301
400B_D008	Comparator 0 configure register (TCON1_COMP0)	32	R/W	0000_0FFFh	58.3.3/3302
400B_D00C	Comparator 1 configure register (TCON1_COMP1)	32	R/W	0000_0FFFh	58.3.4/3303
400B_D010	Comparator 2 configure register (TCON1_COMP2)	32	R/W	0000_0FFFh	58.3.5/3304
400B_D014	Comparator 3 configure register (TCON1_COMP3)	32	R/W	0000_0FFFh	58.3.6/3305
400B_D018	Comparator 0 compare value mask register (TCON1_COMP0_MSK)	32	R/W	0000_0FFFh	58.3.7/3305
400B_D01C	Comparator 1 compare value mask register (TCON1_COMP1_MSK)	32	R/W	0000_0FFFh	58.3.8/3306
400B_D020	Comparator 2 compare value mask register (TCON1_COMP2_MSK)	32	R/W	0000_0FFFh	58.3.9/3306

Table continues on the next page...

TCON memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_D024	Comparator 3 compare value mask register (TCON1_COMP3_MSK)	32	R/W	0000_0FFFh	58.3.10/ 3307
400B_D028	Pulse 0 configure register (TCON1_PULSE0)	32	R/W	0FFF_0FFFh	58.3.11/ 3307
400B_D02C	Pulse 1 configure register (TCON1_PULSE1)	32	R/W	0FFF_0FFFh	58.3.12/ 3308
400B_D030	Pulse 2 configure register (TCON1_PULSE2)	32	R/W	0FFF_0FFFh	58.3.13/ 3310
400B_D034	Pulse 3 configure register (TCON1_PULSE3)	32	R/W	0FFF_0FFFh	58.3.14/ 3311
400B_D038	Pulse 4 configure register (TCON1_PULSE4)	32	R/W	0FFF_0FFFh	58.3.15/ 3312
400B_D03C	Pulse 5 configure register (TCON1_PULSE5)	32	R/W	0FFF_0FFFh	58.3.16/ 3313
400B_D040	Pulse 0 compare value mask register (TCON1_PULSE0_MSK)	32	R/W	0FFF_0FFFh	58.3.17/ 3314
400B_D044	Pulse 1 compare value mask register (TCON1_PULSE1_MSK)	32	R/W	0FFF_0FFFh	58.3.18/ 3315
400B_D048	Pulse 2 compare value mask register (TCON1_PULSE2_MSK)	32	R/W	0FFF_0FFFh	58.3.19/ 3315
400B_D04C	Pulse 3 compare value mask register (TCON1_PULSE3_MSK)	32	R/W	0FFF_0FFFh	58.3.20/ 3316
400B_D050	Pulse 4 compare value mask register (TCON1_PULSE4_MSK)	32	R/W	0FFF_0FFFh	58.3.21/ 3316
400B_D054	Pulse 5 compare value mask register (TCON1_PULSE5_MSK)	32	R/W	0FFF_0FFFh	58.3.22/ 3317
400B_D058	Function control register 0 (TCON1_SMX0)	32	R/W	0000_0000h	58.3.23/ 3318
400B_D05C	Function control register 1 (TCON1_SMX1)	32	R/W	0000_0000h	58.3.24/ 3319
400B_D060	Function control registers 2 (TCON1_SMX2)	32	R/W	0000_0000h	58.3.25/ 3321
400B_D064	Function control register 3 (TCON1_SMX3)	32	R/W	0000_0000h	58.3.26/ 3322
400B_D068	Function control register 4 (TCON1_SMX4)	32	R/W	0000_0000h	58.3.27/ 3324
400B_D06C	Function control register 5 (TCON1_SMX5)	32	R/W	0000_0000h	58.3.28/ 3326
400B_D070	Function control register 6 (TCON1_SMX6)	32	R/W	0000_0000h	58.3.29/ 3327
400B_D074	Function control register 7 (TCON1_SMX7)	32	R/W	0000_0000h	58.3.30/ 3329
400B_D078	Function control register 8 (TCON1_SMX8)	32	R/W	0000_0000h	58.3.31/ 3330

Table continues on the next page...

TCON memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400B_D07C	Function control register 9 (TCON1_SMX9)	32	R/W	0000_0000h	58.3.32/ 3332
400B_D080	Function control register 10 (TCON1_SMX10)	32	R/W	0000_0000h	58.3.33/ 3334
400B_D084	Function control register 11 (TCON1_SMX11)	32	R/W	0000_0000h	58.3.34/ 3335
400B_D088	Function control register 12 (TCON1_SMX12)	32	R/W	0000_0000h	58.3.35/ 3337
400B_D08C	Function control register 13 (TCON1_SMX13)	32	R/W	0000_0000h	58.3.36/ 3338
400B_D090	TCON output mux control low (TCON1_OMUX_LOW)	32	R/W	0000_0000h	58.3.37/ 3340
400B_D094	TCON output mux control high (TCON1_OMUX_HIGH)	32	R/W	0000_0000h	58.3.38/ 3341
400B_D098	TCON look up table 0 (TCON1_LUT0)	32	R/W	FFFF_005Ch	58.3.39/ 3342
400B_D09C	TCON look up table 1 (TCON1_LUT1)	32	R/W	FFFF_005Ch	58.3.40/ 3343
400B_D0A0	TCON look up table 2 (TCON1_LUT2)	32	R/W	FFFF_005Ch	58.3.41/ 3343
400B_D0A4	TCON look up table 3 (TCON1_LUT3)	32	R/W	FFFF_005Ch	58.3.42/ 3343
400B_D0A8	TCON look up table 4 (TCON1_LUT4)	32	R/W	FFFF_005Ch	58.3.43/ 3344
400B_D0AC	TCON look up table 5 (TCON1_LUT5)	32	R/W	FFFF_005Ch	58.3.44/ 3344
400B_D0B0	TCON look up table 6 (TCON1_LUT6)	32	R/W	FFFF_005Ch	58.3.45/ 3344
400B_D0B4	TCON look up table 7 (TCON1_LUT7)	32	R/W	FFFF_005Ch	58.3.46/ 3345
400B_D0B8	TCON look up table 8 (TCON1_LUT8)	32	R/W	FFFF_005Ch	58.3.47/ 3345
400B_D0BC	TCON look up tables 9 (TCON1_LUT9)	32	R/W	FFFF_005Ch	58.3.48/ 3345
400B_D0C0	TCON look up table 10 (TCON1_LUT10)	32	R/W	FFFF_005Ch	58.3.49/ 3346
400B_D0C4	TCON look up table 11 (TCON1_LUT11)	32	R/W	FFFF_005Ch	58.3.50/ 3346
400B_D0C8	TCON look up table 12 (TCON1_LUT12)	32	R/W	FFFF_005Ch	58.3.51/ 3346
400B_D0CC	TCON look up table 13 (TCON1_LUT13)	32	R/W	FFFF_005Ch	58.3.52/ 3347
400B_D104	TCON control2 register (TCON1_CTRL2)	32	R/W	0000_0002h	58.3.53/ 3347

58.3.1 TCON control1 register (TCONx_CTRL1)

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	TCON_EN	Reserved	TCON_BYPASS	INV_EN	TCONx_INV											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	INIT_DELAY			V_REF_SEL			H_REF_SEL		VLEN		HSYNC_INV	VSYNC_INV	COLOR_DEPTH	RGB_PADDING_EN	RGB_PADDING	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

TCONx_CTRL1 field descriptions

Field	Description
31 TCON_EN	**NOTE:** Program the TCON_CTRL1 register to configure the operation mode of TCON and enable TCON (It is recommended to do this in two steps, and set TCON_EN in the second). 0 Disable TCON. 1 enable TCON.
30 RESERVED	This field is reserved.
29 TCON_BYPASS	Controls whether TCON is bypassed. 0 Not bypass TCON. State of the TCON is decided by TCON_EN. 1 Bypass TCON, both data and timing signals will pass through the TCON unmodified. see the "Bypass Mode" section for pin mapping in that mode.
28 INV_EN	The input pixel (n) is compared with pixel (n-1). If more than half of the RGB data bits toggle, the pixel (n) output will be inverted and the tcon_out[11] flags this data inversion to the gray scale reference. tcon_out[11] =1 -> data_out is inverted, tcou_out[11] =0 -> data_out is not inverted. Intention of this feature is for board level EMI reduction. 0 Disable output data inversion 1 Enable output data inversion

Table continues on the next page...

TCONx_CTRL1 field descriptions (continued)

Field	Description
27–16 TCONx_INV	TCONx output inversion control 0 do not invert tconx 1 invert output tconx
15–13 INIT_DELAY	(INIT_DELAY[2:0]) Initialization delay, set the number of frames before the TCON starts to output timing signals.
12–10 V_REF_SEL	(V_REF_SEL[2:0]) v_ref selector, please refer to the "Toggle generator" section for use of v_ref signal. Note: pulse generated must be vertical based for the toggle generator. 000 select tcon_pulse0 output as v_ref 001 select tcon_pulse1 output as v_ref 010 select tcon_pulse2 output as v_ref 011 select tcon_pulse3 output as v_ref 100 select tcon_pulse4 output as v_ref 101 select tcon_pulse5 output as v_ref 110 select tcon_pulse0 output as v_ref, while reset vtgl_counter and vtgl[3] to 0 at rising edge of v_ref. 111 select tcon_pulse0 output as v_ref, while set vtgl_counter to 1 and reset vtgl[3] to 0 at rising edge of v_ref.
9–8 H_REF_SEL	(H_REF_SEL[2:0]) h_ref selector, please refer to the "Toggle generator" section for use of h_ref signal. 00 select tcon_comp0 output as h_ref. 01 select tcon_comp1 output as h_ref. 10 select tcon_comp2 output as h_ref. 11 select tcon_comp3 output as h_ref.
7–6 VLEN	(VLEN[1:0]) Maximum counter value for vtgl_counter in the toggle generator. Please refer to to the "Toggle generator" section for detailed functionality of this field.
5 HSYNC_INV	Horizontal sync. 0 hsync_in signal is active high 1 hsync_in signal is active low
4 VSYNC_INV	Vertical sync. 0 vsync_in signal is active high 1 vsync_in signal is active low
3 COLOR_DEPTH	Color depth of each color component 0 6 bits, the 2 LSB's are set to 2'b0 1 8 bits
2 RGB_PADDING_EN	RGB data padding enable 0 disable padding during blanking 1 enable padding during blanking
1 RGB_PADDING	RGB data driven during blanking. 0 all "0" 1 all "1"
0 RESERVED	This field is reserved.

58.3.2 Bit map control register (TCONx_BMC)

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						CLK_POS					COLOR_ORDER			BIT_ORDER	Reserved
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_BMC field descriptions

Field	Description
31–10 RESERVED	This field is reserved.
9–5 CLK_POS	(CLK_POS[4:0]) Clock position selection, output pixel clock can be assigned to any data_out pins. Please refer to the "Clock mapping in TTL mode" table for detailed mapping relationship.
4–2 COLOR_ORDER	(COLOR_ORDER[2:0]) Color component order configuration bits. Other values are reserved. 000 RGB 001 BRG 010 GBR 011 RBG 100 GRB 101 BGR
1 BIT_ORDER	Controls the bit order. 0 MSB 7 down to LSB 0 for every color component 1 LSB 0 up to MSB 7, for every color component. (inverted order)
0 RESERVED	This field is reserved.

58.3.3 Comparator 0 configure register (TCONx_COMP0)

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	FUNC_SEL	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved				COMP_VALUE											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_COMP0 field descriptions

Field	Description
31 FUNC_SEL	On which direction the comparation is based, vertical or horizontal. 0 compare on horizontal direction 1 compare on vertical direction
30–12 RESERVED	This field is reserved.
11–0 COMP_VALUE	(COMP_VALUE[11:0]) Comparison value. FUNC_SEL=0 -> An one pixel clock cycle high pulse is generated when h_count = COMP_VALUE + 2; FUNC_SEL=1 -> An one line wide high pulse is generated when v_count = COMP_VALUE. See the "Comparator" section for a detailed explanation.

58.3.4 Comparator 1 configure register (TCONx_COMP1)

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	FUNC_SEL	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved				COMP_VALUE											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_COMP1 field descriptions

Field	Description
31 FUNC_SEL	On which direction the comparation is based, vertical or horizontal. 0 compare on horizontal direction 1 compare on vertical direction
30–12 RESERVED	This field is reserved.
11–0 COMP_VALUE	(COMP_VALUE[11:0]) Comparison value. FUNC_SEL=0 -> An one pixel clock cycle high pulse is generated when h_count = COMP_VALUE + 2; FUNC_SEL=1 -> An one line wide high pulse is generated when v_count = COMP_VALUE. See the "Comparator" section for a detailed explanation.

58.3.5 Comparator 2 configure register (TCONx_COMP2)

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	FUNC_SEL	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved				COMP_VALUE											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_COMP2 field descriptions

Field	Description
31 FUNC_SEL	On which direction the comparation is based, vertical or horizontal. 0 compare on horizontal direction 1 compare on vertical direction
30–12 RESERVED	This field is reserved.
11–0 COMP_VALUE	(COMP_VALUE[11:0]) Comparison value. FUNC_SEL=0 -> An one pixel clock cycle high pulse is generated when h_count = COMP_VALUE + 2; FUNC_SEL=1 -> An one line wide high pulse is generated when v_count = COMP_VALUE. See the "Comparator" section for a detailed explanation.

58.3.6 Comparator 3 configure register (TCONx_COMP3)

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	FUNC_SEL	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved				COMP_VALUE											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_COMP3 field descriptions

Field	Description
31 FUNC_SEL	On which direction the comparation is based, vertical or horizontal. 0 compare on horizontal direction 1 compare on vertical direction
30–12 RESERVED	This field is reserved.
11–0 COMP_VALUE	(COMP_VALUE[11:0]) Comparison value. FUNC_SEL=0 -> An one pixel clock cycle high pulse is generated when h_count = COMP_VALUE + 2; FUNC_SEL=1 -> An one line wide high pulse is generated when v_count = COMP_VALUE. See the "Comparator" section for a detailed explanation.

58.3.7 Comparator 0 compare value mask register (TCONx_COMP0_MSK)

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved																				MSK											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_COMP0_MSK field descriptions

Field	Description
31–12 RESERVED	This field is reserved.

Table continues on the next page...

TCONx_COMP0_MSK field descriptions (continued)

Field	Description
11–0 MSK	(MSK[11:0]) Comparison value mask. For a given MSK[x]: 0 Ignore the given bit x in comparator matching 1 Include the given bit x in comparator matching

58.3.8 Comparator 1 compare value mask register (TCONx_COMP1_MSK)

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																				MSK											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_COMP1_MSK field descriptions

Field	Description
31–12 RESERVED	This field is reserved.
11–0 MSK	(MSK[11:0]) Comparison value mask. For a given MSK[x]: 0 Ignore the given bit x in comparator matching 1 Include the given bit x in comparator matching

58.3.9 Comparator 2 compare value mask register (TCONx_COMP2_MSK)

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																				MSK											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_COMP2_MSK field descriptions

Field	Description
31–12 RESERVED	This field is reserved.
11–0 MSK	(MSK[11:0]) Comparison value mask. For a given MSK[x]: 0 Ignore the given bit x in comparator matching 1 Include the given bit x in comparator matching

58.3.10 Comparator 3 compare value mask register (TCONx_COMP3_MSK)

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																				MSK											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_COMP3_MSK field descriptions

Field	Description
31–12 RESERVED	This field is reserved.
11–0 MSK	(MSK[11:0]) Comparison value mask. For a given MSK[x]: 0 Ignore the given bit x in comparator matching 1 Include the given bit x in comparator matching

58.3.11 Pulse 0 configure register (TCONx_PULSE0)

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FUNC_SEL		Reserved		SET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMPARATOR_SEL		Reserved		RESET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE0 field descriptions

Field	Description
31–30 FUNC_SEL	(FUNC_SEL[1:0]) Select the type of pulse, horizontal/vertical/or mix 00 SET/RESET value both for horizontal compare 01 SET/RESET value both for vertical compare, signal transition point determined by COMPARATOR_SEL

Table continues on the next page...

TCONx_PULSE0 field descriptions (continued)

Field	Description
	10 SET/RESET value both for vertical compare, signal transitions at the beginning of the line which is matched. 11 SET value for horizontal compare, RESET value for vertical compare. Signal transition point for RESET determined by COMPARATOR_SEL.
29–28 RESERVED	This field is reserved.
27–16 SET	(SET[11:0]) Set point compare value Note: There will be a two pixel clock cycle delay between the internal pixel counter and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.
15–14 COMPARATOR_SEL	(COMPARATOR_SEL[1:0) When FUNC_SEL is set to 01, 1 of the 4 comparator outputs is selected to define the horizontal change point for both SET/RESET. When FUNC_SEL is set to 11, 1 of the 4 comparator outputs is selected to define the horizontal change point for RESET. Note: when FUNC_SEL set to 01 or 11, user should program the corresponding comparator for horizontal compare to ensure the timing signal is generated as expected. 00 select comparator0 01 select comparator1 10 select comparator2 11 select comparator3
13–12 RESERVED	This field is reserved.
11–0 RESET	(RESET[11:0]) Reset point compare value Note: There will be a two pixel clock cycle's delay between the internal pixel count and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.

58.3.12 Pulse 1 configure register (TCONx_PULSE1)

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	FUNC_SEL		Reserved		SET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	COMPARATOR_SEL		Reserved		RESET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE1 field descriptions

Field	Description
31–30 FUNC_SEL	(FUNC_SEL[1:0]) Select the type of pulse, horizontal/vertical/or mix 00 SET/RESET value both for horizontal compare 01 SET/RESET value both for vertical compare, signal transition point determined by COMPARATOR_SEL 10 SET/RESET value both for vertical compare, signal transitions at the beginning of the line which is matched. 11 SET value for horizontal compare, RESET value for vertical compare. Signal transition point for RESET determined by COMPARATOR_SEL.
29–28 RESERVED	This field is reserved.
27–16 SET	(SET[11:0]) Set point compare value Note: There will be a two pixel clock cycle delay between the internal pixel counter and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.
15–14 COMPARATOR_ SEL	(COMPARATOR_SEL[1:0) When FUNC_SEL is set to 01, 1 of the 4 comparator outputs is selected to define the horizontal change point for both SET/RESET. When FUNC_SEL is set to 11, 1 of the 4 comparator outputs is selected to define the horizontal change point for RESET. Note: when FUNC_SEL set to 01 or 11, user should program the corresponding comparator for horizontal compare to ensure the timing signal is generated as expected. 00 select comparator0 01 select comparator1 10 select comparator2 11 select comparator3
13–12 RESERVED	This field is reserved.
11–0 RESET	(RESET[11:0]) Reset point compare value Note: There will be a two pixel clock cycle's delay between the internal pixel count and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.

58.3.13 Pulse 2 configure register (TCONx_PULSE2)

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	FUNC_SEL		Reserved		SET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	COMPARATOR_SEL		Reserved		RESET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE2 field descriptions

Field	Description
31–30 FUNC_SEL	(FUNC_SEL[1:0]) Select the type of pulse, horizontal/vertical/or mix 00 SET/RESET value both for horizontal compare 01 SET/RESET value both for vertical compare, signal transition point determined by COMPARATOR_SEL 10 SET/RESET value both for vertical compare, signal transitions at the beginning of the line which is matched. 11 SET value for horizontal compare, RESET value for vertical compare. Signal transition point for RESET determined by COMPARATOR_SEL.
29–28 RESERVED	This field is reserved.
27–16 SET	(SET[11:0]) Set point compare value Note: There will be a two pixel clock cycle delay between the internal pixel counter and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.
15–14 COMPARATOR_ SEL	(COMPARATOR_SEL[1:0) When FUNC_SEL is set to 01, 1 of the 4 comparator outputs is selected to define the horizontal change point for both SET/RESET. When FUNC_SEL is set to 11, 1 of the 4 comparator outputs is selected to define the horizontal change point for RESET. Note: when FUNC_SEL set to 01 or 11, user should program the corresponding comparator for horizontal compare to ensure the timing signal is generated as expected. 00 select comparator0 01 select comparator1 10 select comparator2 11 select comparator3

Table continues on the next page...

TCONx_PULSE2 field descriptions (continued)

Field	Description
13–12 RESERVED	This field is reserved.
11–0 RESET	(RESET[11:0]) Reset point compare value Note: There will be a two pixel clock cycle's delay between the internal pixel count and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.

58.3.14 Pulse 3 configure register (TCONx_PULSE3)

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	FUNC_SEL		Reserved		SET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	COMPARATOR_SEL		Reserved		RESET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE3 field descriptions

Field	Description
31–30 FUNC_SEL	(FUNC_SEL[1:0]) Select the type of pulse, horizontal/vertical/or mix 00 SET/RESET value both for horizontal compare 01 SET/RESET value both for vertical compare, signal transition point determined by COMPARATOR_SEL 10 SET/RESET value both for vertical compare, signal transitions at the beginning of the line which is matched. 11 SET value for horizontal compare, RESET value for vertical compare. Signal transition point for RESET determined by COMPARATOR_SEL.
29–28 RESERVED	This field is reserved.
27–16 SET	(SET[11:0]) Set point compare value Note: There will be a two pixel clock cycle delay between the internal pixel counter and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.

Table continues on the next page...

TCONx_PULSE3 field descriptions (continued)

Field	Description
15–14 COMPARATOR_SEL	(COMPARATOR_SEL[1:0) When FUNC_SEL is set to 01, 1 of the 4 comparator outputs is selected to define the horizontal change point for both SET/RESET. When FUNC_SEL is set to 11, 1 of the 4 comparator outputs is selected to define the horizontal change point for RESET. Note: when FUNC_SEL set to 01 or 11, user should program the corresponding comparator for horizontal compare to ensure the timing signal is generated as expected. 00 select comparator0 01 select comparator1 10 select comparator2 11 select comparator3
13–12 RESERVED	This field is reserved.
11–0 RESET	(RESET[11:0]) Reset point compare value Note: There will be a two pixel clock cycle's delay between the internal pixel count and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.

58.3.15 Pulse 4 configure register (TCONx_PULSE4)

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	FUNC_SEL		Reserved		SET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	COMPARATOR_SEL		Reserved		RESET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE4 field descriptions

Field	Description
31–30 FUNC_SEL	(FUNC_SEL[1:0]) Select the type of pulse, horizontal/vertical/or mix 00 SET/RESET value both for horizontal compare 01 SET/RESET value both for vertical compare, signal transition point determined by COMPARATOR_SEL

Table continues on the next page...

TCONx_PULSE4 field descriptions (continued)

Field	Description
	10 SET/RESET value both for vertical compare, signal transitions at the beginning of the line which is matched. 11 SET value for horizontal compare, RESET value for vertical compare. Signal transition point for RESET determined by COMPARATOR_SEL.
29–28 RESERVED	This field is reserved.
27–16 SET	(SET[11:0]) Set point compare value Note: There will be a two pixel clock cycle delay between the internal pixel counter and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.
15–14 COMPARATOR_ SEL	(COMPARATOR_SEL[1:0) When FUNC_SEL is set to 01, 1 of the 4 comparator outputs is selected to define the horizontal change point for both SET/RESET. When FUNC_SEL is set to 11, 1 of the 4 comparator outputs is selected to define the horizontal change point for RESET. Note: when FUNC_SEL set to 01 or 11, user should program the corresponding comparator for horizontal compare to ensure the timing signal is generated as expected. 00 select comparator0 01 select comparator1 10 select comparator2 11 select comparator3
13–12 RESERVED	This field is reserved.
11–0 RESET	(RESET[11:0]) Reset point compare value Note: There will be a two pixel clock cycle's delay between the internal pixel count and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.

58.3.16 Pulse 5 configure register (TCONx_PULSE5)

Address: Base address + 3Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	FUNC_SEL		Reserved		SET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	COMPARATOR_SEL		Reserved		RESET											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE5 field descriptions

Field	Description
31–30 FUNC_SEL	(FUNC_SEL[1:0]) Select the type of pulse, horizontal/vertical/or mix 00 SET/RESET value both for horizontal compare 01 SET/RESET value both for vertical compare, signal transition point determined by COMPARATOR_SEL 10 SET/RESET value both for vertical compare, signal transitions at the beginning of the line which is matched. 11 SET value for horizontal compare, RESET value for vertical compare. Signal transition point for RESET determined by COMPARATOR_SEL.
29–28 RESERVED	This field is reserved.
27–16 SET	(SET[11:0]) Set point compare value Note: There will be a two pixel clock cycle delay between the internal pixel counter and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.
15–14 COMPARATOR_SEL	(COMPARATOR_SEL[1:0) When FUNC_SEL is set to 01, 1 of the 4 comparator outputs is selected to define the horizontal change point for both SET/RESET. When FUNC_SEL is set to 11, 1 of the 4 comparator outputs is selected to define the horizontal change point for RESET. Note: when FUNC_SEL set to 01 or 11, user should program the corresponding comparator for horizontal compare to ensure the timing signal is generated as expected. 00 select comparator0 01 select comparator1 10 select comparator2 11 select comparator3
13–12 RESERVED	This field is reserved.
11–0 RESET	(RESET[11:0]) Reset point compare value Note: There will be a two pixel clock cycle's delay between the internal pixel count and the output pixel data in horizontal direction, user should take this two cycle delay into account when programming this field. See the "Pulse generator" section for a detailed explanation.

58.3.17 Pulse 0 compare value mask register (TCONx_PULSE0_MSK)

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved				SET_MSK											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved				RESET_MSK											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE0_MSK field descriptions

Field	Description
31–28 RESERVED	This field is reserved.

Table continues on the next page...

TCONx_PULSE0_MSK field descriptions (continued)

Field	Description
27–16 SET_MSK	(SET_MSK[11:0]) Set point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator set value comparison 1 Include x postion in pulse generator set value comparison
15–12 RESERVED	This field is reserved.
11–0 RESET_MSK	(RESET_MSK[11:0]) Reset point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator reset value comparison. 1 Include x position in pulse generator reset value comparison

58.3.18 Pulse 1 compare value mask register (TCONx_PULSE1_MSK)

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SET_MSK												Reserved				RESET_MSK											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE1_MSK field descriptions

Field	Description
31–28 RESERVED	This field is reserved.
27–16 SET_MSK	(SET_MSK[11:0]) Set point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator set value comparison 1 Include x postion in pulse generator set value comparison
15–12 RESERVED	This field is reserved.
11–0 RESET_MSK	(RESET_MSK[11:0]) Reset point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator reset value comparison. 1 Include x position in pulse generator reset value comparison

58.3.19 Pulse 2 compare value mask register (TCONx_PULSE2_MSK)

Address: Base address + 48h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SET_MSK												Reserved				RESET_MSK											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE2_MSK field descriptions

Field	Description
31–28 RESERVED	This field is reserved.
27–16 SET_MSK	(SET_MSK[11:0]) Set point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator set value comparison 1 Include x postion in pulse generator set value comparison
15–12 RESERVED	This field is reserved.
11–0 RESET_MSK	(RESET_MSK[11:0]) Reset point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator reset value comparison. 1 Include x position in pulse generator reset value comparison

58.3.20 Pulse 3 compare value mask register (TCONx_PULSE3_MSK)

Address: Base address + 4Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SET_MSK												Reserved				RESET_MSK											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE3_MSK field descriptions

Field	Description
31–28 RESERVED	This field is reserved.
27–16 SET_MSK	(SET_MSK[11:0]) Set point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator set value comparison 1 Include x postion in pulse generator set value comparison
15–12 RESERVED	This field is reserved.
11–0 RESET_MSK	(RESET_MSK[11:0]) Reset point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator reset value comparison. 1 Include x position in pulse generator reset value comparison

58.3.21 Pulse 4 compare value mask register (TCONx_PULSE4_MSK)

Address: Base address + 50h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SET_MSK												Reserved				RESET_MSK											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE4_MSK field descriptions

Field	Description
31–28 RESERVED	This field is reserved.
27–16 SET_MSK	(SET_MSK[11:0]) Set point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator set value comparison 1 Include x postion in pulse generator set value comparison
15–12 RESERVED	This field is reserved.
11–0 RESET_MSK	(RESET_MSK[11:0]) Reset point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator reset value comparison. 1 Include x position in pulse generator reset value comparison

58.3.22 Pulse 5 compare value mask register (TCONx_PULSE5_MSK)

Address: Base address + 54h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				SET_MSK												Reserved				RESET_MSK											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

TCONx_PULSE5_MSK field descriptions

Field	Description
31–28 RESERVED	This field is reserved.
27–16 SET_MSK	(SET_MSK[11:0]) Set point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator set value comparison 1 Include x postion in pulse generator set value comparison
15–12 RESERVED	This field is reserved.
11–0 RESET_MSK	(RESET_MSK[11:0]) Reset point compare mask value. For SET_MSK[x] value: 0 Ignore x position in pulse generator reset value comparison. 1 Include x position in pulse generator reset value comparison

58.3.23 Function control register 0 (TCONx_SMX0)

Address: Base address + 58h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX0 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2]

Table continues on the next page...

TCONx_SMX0 field descriptions (continued)

Field	Description
	0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.24 Function control register 1 (TCONx_SMX1)

Address: Base address + 5Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX1 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y;

Table continues on the next page...

TCONx_SMX1 field descriptions (continued)

Field	Description
	100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1

Table continues on the next page...

TCONx_SMX1 field descriptions (continued)

Field	Description
	1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.25 Function control registers 2 (TCONx_SMX2)

Address: Base address + 60h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX2 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1

Table continues on the next page...

TCONx_SMX2 field descriptions (continued)

Field	Description
	04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.26 Function control register 3 (TCONx_SMX3)

Address: Base address + 64h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX3 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8

Table continues on the next page...

TCONx_SMX3 field descriptions (continued)

Field	Description
	15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.27 Function control register 4 (TCONx_SMX4)

Address: Base address + 68h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX4 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.

Table continues on the next page...

TCONx_SMX4 field descriptions (continued)

Field	Description
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.28 Function control register 5 (TCONx_SMX5)

Address: Base address + 6Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX5 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2]

Table continues on the next page...

TCONx_SMX5 field descriptions (continued)

Field	Description
	0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.29 Function control register 6 (TCONx_SMX6)

Address: Base address + 70h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX6 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y;

Table continues on the next page...

TCONx_SMX6 field descriptions (continued)

Field	Description
	100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1

Table continues on the next page...

TCONx_SMX6 field descriptions (continued)

Field	Description
	1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.30 Function control register 7 (TCONx_SMX7)

Address: Base address + 74h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX7 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1

Table continues on the next page...

TCONx_SMX7 field descriptions (continued)

Field	Description
	04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.31 Function control register 8 (TCONx_SMX8)

Address: Base address + 78h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX8 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8

Table continues on the next page...

TCONx_SMX8 field descriptions (continued)

Field	Description
	15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.32 Function control register 9 (TCONx_SMX9)

Address: Base address + 7Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX9 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.

Table continues on the next page...

TCONx_SMX9 field descriptions (continued)

Field	Description
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.33 Function control register 10 (TCONx_SMX10)

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX10 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2]

Table continues on the next page...

TCONx_SMX10 field descriptions (continued)

Field	Description
	0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.34 Function control register 11 (TCONx_SMX11)

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX11 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y;

Table continues on the next page...

TCONx_SMX11 field descriptions (continued)

Field	Description
	100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1

Table continues on the next page...

TCONx_SMX11 field descriptions (continued)

Field	Description
	1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.35 Function control register 12 (TCONx_SMX12)

Address: Base address + 88h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX12 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1

Table continues on the next page...

TCONx_SMX12 field descriptions (continued)

Field	Description
	04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8 15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.36 Function control register 13 (TCONx_SMX13)

Address: Base address + 8Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INDEX3_SEL			INDEX2_SEL			INDEX1_SEL			INDEX0_SEL			Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						Y_SEL					X_SEL				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_SMX13 field descriptions

Field	Description
31–29 INDEX3_SEL	(INDEX3_SEL[2:0]) SMXx LUT index3 selection. {index3,index2,index1,index0} will be used as address to LUT to generate timing signal.} 000 index3 = 0; 001 index3 = X; 010 index3 = Y; 011 index3 = X&Y; 100 index3 = X|Y; 101 index3 = X^Y; 110 index3 = !(X&Y)) 111 reserved for use
28–26 INDEX2_SEL	(INDEX2_SEL[2:0]) SMXx LUT index2 selection. Same functionality as INDEX3_SEL
25–23 INDEX1_SEL	(INDEX1_SEL[2:0]) SMXx LUT index1 selection. Same functionality as INDEX3_SEL
22–20 INDEX0_SEL	(INDEX0_SEL[2:0]) SMXx LUT index0 selection. Same functionality as INDEX3_SEL
19–10 RESERVED	This field is reserved.
9–5 Y_SEL	(Y_SEL[4:0]) SMXx logic input Y selection. 1C-1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX0 0D: SMX1 0E: SMX2 0F: SMX3 10: SMX4 11: SMX5 12: SMX6 13: SMX7 14: SMX8

Table continues on the next page...

TCONx_SMX13 field descriptions (continued)

Field	Description
	15: SMX9 16: SMX10 17: SMX11 18: comp0 19: comp1 1A: comp2 1B: comp3
4–0 X_SEL	(X_SEL[4:0]) SMX logic input A selection, the same selection logic is implemented here as in Y_SEL.

58.3.37 TCON output mux control low (TCONx_OMUX_LOW)

Address: Base address + 90h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		TCON5					TCON4					TCON3			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCON3	TCON2					TCON1					TCON0				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_OMUX_LOW field descriptions

Field	Description
31–30 RESERVED	This field is reserved.
29–25 TCON5	(TCON5[4:0]) Output selection for tcon pin5, refer to field TCON0 for detailed functional description
24–20 TCON4	(TCON4[4:0]) Output selection for tcon pin4, refer to field TCON0 for detailed functional description
19–15 TCON3	(TCON3[4:0]) Output selection for tcon pin3, refer to field TCON0 for detailed functional description
14–10 TCON2	(TCON2[4:0]) Output selection for tcon pin2, refer to field TCON0 for detailed functional description
9–5 TCON1	(TCON1[4:0]) Output selection for tcon pin1, refer to field TCON0 for detailed functional description
4–0 TCON0	(TCON0[4:0]) Output selection for tcon pin0. 14 - 1F are reserved, default to const0. 00: const0 01: const1 02: pulse0

Table continues on the next page...

TCONx_OMUX_LOW field descriptions (continued)

Field	Description
	03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX6 0D: SMX7 0E: SMX8 0F: SMX9 10: SMX10 11: SMX11 12: SMX12 13: SMX13

58.3.38 TCON output mux control high (TCONx_OMUX_HIGH)

Address: Base address + 94h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		TCON11					TCON10					TCON9			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCON9	TCON8					TCON7					TCON6				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCONx_OMUX_HIGH field descriptions

Field	Description
31–30 RESERVED	This field is reserved.
29–25 TCON11	(TCON11[4:0]) Output selection for tcon pin11, refer to field TCON6 for detailed functional description
24–20 TCON10	(TCON10[4:0]) Output selection for tcon pin10, refer to field TCON6 for detailed functional description

Table continues on the next page...

TCONx_OMUX_HIGH field descriptions (continued)

Field	Description
19–15 TCON9	(TCON9[4:0]) Output selection for tcon pin9, refer to field TCON6 for detailed functional description
14–10 TCON8	(TCON8[4:0]) Output selection for tcon pin8, refer to field TCON6 for detailed functional description
9–5 TCON7	(TCON7[4:0]) Output selection for tcon pin7, refer to field TCON6 for detailed functional description
4–0 TCON6	(TCON6[4:0]) Output selection for tcon pin6. 14 - 1F are reserved values. 00: const0 01: const1 02: pulse0 03: pulse1 04: pulse2 05: pulse3 06: pulse4 07: pulse5 08: vtgl[0] 09: vtgl[1] 0A: vtgl[2] 0B: vtgl[3] 0C: SMX6 0D: SMX7 0E: SMX8 0F: SMX9 10: SMX10 11: SMX11 12: SMX12 13: SMX13

58.3.39 TCON look up table 0 (TCONx_LUT0)

Address: Base address + 98h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT0 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx.

TCONx_LUT0 field descriptions (continued)

Field	Description
	SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.40 TCON look up table 1 (TCONx_LUT1)

Address: Base address + 9Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT1 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.41 TCON look up table 2 (TCONx_LUT2)

Address: Base address + A0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT2 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.42 TCON look up table 3 (TCONx_LUT3)

Address: Base address + A4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT3 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.43 TCON look up table 4 (TCONx_LUT4)

Address: Base address + A8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT4 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.44 TCON look up table 5 (TCONx_LUT5)

Address: Base address + ACh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT5 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.45 TCON look up table 6 (TCONx_LUT6)

Address: Base address + B0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT6 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.46 TCON look up table 7 (TCONx_LUT7)

Address: Base address + B4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT7 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.47 TCON look up table 8 (TCONx_LUT8)

Address: Base address + B8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT8 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.48 TCON look up tables 9 (TCONx_LUT9)

Address: Base address + BCh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT9 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.49 TCON look up table 10 (TCONx_LUT10)

Address: Base address + C0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT10 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.50 TCON look up table 11 (TCONx_LUT11)

Address: Base address + C4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT11 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.51 TCON look up table 12 (TCONx_LUT12)

Address: Base address + C8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT12 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.52 TCON look up table 13 (TCONx_LUT13)

Address: Base address + CCh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LUT																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

TCONx_LUT13 field descriptions

Field	Description
31–0 LUT	(LUT[31:0]) Look up table for SMXx. SMXx(n) = TCON_LUTx[{index3,index2,index1,index0,SMXx(n-1)}], see the "TCON_SMXx Register Field Descriptions" for construction of index3 to index0.

58.3.53 TCON control2 register (TCONx_CTRL2)

Address: Base address + 104h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							CLK_OFFSET									Reserved								DIV_RATIO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

TCONx_CTRL2 field descriptions

Field	Description
31–25 RESERVED	This field is reserved.
24–16 CLK_OFFSET	(CLK_OFFSET[8:0]) Output pixel clock offset value in half ipg_clk cycle unit. please refer to the "Clock/ Data Skew Adjustment" section for use of this field. The valid range of this field is to 0 to 2N-1 in TTL mode, where N=(DIV_RATIO+1).
15–8 RESERVED	This field is reserved.
7–0 DIV_RATIO	(DIV_RATIO[7:0]) TCON pixel clock divide ratio. When DIV_RATIO set to N, pix_clk will be ipg_clk divided by N+1.

58.4 Functional Description

The major functions of the TCON are to generate timing signals needed to drive the external row drivers and column drivers, and map the RGB data bits to output pins. It accepts RGB data, Horizontal/Vertical synchronization signals from the Display Controller Unit module, maintains two internal counters (x_count, and y_count), and uses these to generate the timing signals. In order to support as many types of panels as possible, the timing generation unit has been designed to be flexible and can be configured freely to generate complicated timing signals. See Timing signal generator for details about the timing signal generation unit.

The TCON has two operation modes: TTL mode, and bypass mode. In bypass mode the input signals are passed through the TCON unchanged, and in TTL mode the TCON performs signal (the RGB data and pixel clock) bit mapping. For details of the operation modes see Modes of operation, and for the signal bit mapping see Bit Mapping Control (BMC).

58.4.1 Modes of operation

Various operation modes are discussed below.

58.4.1.1 TTL mode

In TTL mode the TCON drives the RGB data, pixel clock, and the TCON timing signals all via TTL interface. In this mode simple signal bit mapping is performed, see Bit mapping in TTL mode.

It is also possible to assign the pixel clock to any one of the 25 output pins in TTL mode, see Clock mapping in TTL mode.

To enable the TTL mode, the user shall set the TCON_CTRL1[TCON_BYPASS]='0', and TCON_CTRL1[TCON_EN]='1'.

58.4.1.2 Bypass mode

For applications where the TCON is not in use, a bypass mode is provided in which the input signals from the Display Control Unit are passed through the TCON unchanged. See the table below for how the input signals are assigned to the TCON output.

Table 58-164. input/output signal mapping in bypass mode

Input	Output
data_in[23:0]	data_out[25:2]
pix_clk	data_out[1]
dcu_tag	tcon_out[0]
hsync_in	tcon_out[1]
vsync_in	tcon_out[2]
data_en_in	tcon_out[3]

To put the TCON in bypass mode set TCON_CTRL1[TCON_EN]='0' and TCON_CTRL1[TCON_BYPASS]='1'.

58.4.2 Timing signal generator

The Timing Signal Generator (TSG) is the TCON timing signal generation unit. It accepts the HSYNC/VSYNC signals from the input parallel display interface, and maintains two counters (*hcount, vcount*) based on that. All the timing signals are generated based on these two counters.

For every line the *hcount* counts the current pixel number, from 1 to the number of pixels per line (the full line, including inactive period). It restarts from 1 with the HSYNC signal rising edge.

Similarly the *vcount* counts from 1 with the VSYNC signal rising edge (first line), increments by "1" at the end of each line, until the last line in a frame (including inactive lines).

The timing signal generator is composed of 4 comparators (Comparator), 6 pulse generators (Pulse generator), 1 toggle generator (Toggle generator), and a signal mixer or synthesizer (Signal Mixer (SMX)). See the following sections for detailed descriptions.

58.4.2.1 Comparator

There are 4 comparators in the TCON, each can be configured to compare *hcount* or *vcount* with the value specified in a TCON_COMPx Register. If desired the user can mask some bits from comparing by setting the corresponding bits in both TCON_COMPx_MSK and TCON_COMPx register to "0". A one-cycle (pixel clock cycle, or line cycle) pulse will be generated when the compare result is match, and the comparison logic is shown below:

$$compare\_out = (TCON\_COMPx.COMP\_VALUE == (TCON\_COMPx\_MSK.MASK \ \& \ hcount))$$

or

$$compare\_out = (TCON\_COMPx.COMP\_VALUE == (TCON\_COMPx\_MSK.MASK \,\&\, vcount))$$

TCON_COMPx[FUNC_SEL] determines which of the above equations is used (i.e. comparison is done on horizontal or vertical direction).

58.4.2.2 Pulse generator

There are 6 pulse generators in the TCON which can be used to generate pulses which have a *set* point and a *reset* point, and the pulse length is discretional. The *set* and *reset* point is determined by TCON_PULSEx[SET] and TCON_PULSEx[RESET], to which the *hcount* or *vcount* value will be compared. If desired the user can mask some bits from comparing by setting the corresponding bits in both TCON_PULSEx_MSK and TCON_PULSEx register to 0. The equations used to find the *set/clear* point are similar to the equations given in the "Comparator" section above.

TCON_PULSEx[FUNC_SEL] controls the type of the pulse, a.k.a. whether the *set/reset* point comparison is performed on horizontal direction (compare with *hcount*) or vertical direction (compare with *vcount*). And when vertical comparison is selected (ie. FUNC_SEL = 01 or 11), 1 of the 4 comparator outputs (must be a horizontal pulse) can be selected to further determine the signal transition point on horizontal direction. Or when FUNC_SEL = 10, the signal transition will happen immediately when the vertical compare matches (i.e. at the beginning of the line). When needed TCON_PULSEx[COMPARATOR_SEL] selects 1 of the 4 comparator outputs as the horizontal reference.

58.4.2.3 Toggle generator

There is a toggle generator in the TCON that can be used to generate signals which toggles line to line, or frame to frame, or signal which toggles line to line but polarity changes from frame to frame. The toggle generator accepts the 4 comparator outputs (TCON_COMP[0:3]) and the 6 pulses (TCON_PULSE[0-5]), and generates 4 toggle signals (vtgl*[0-3]*) based on that.

The toggle generator uses a counter (vtgl_counter) and a T flip-flop to generate the 4 toggle signals. The vtgl_counter takes 1 of the 4 comparator outputs as the clock (*h_ref*), and 1 of the 6 pulses as the enable signal (*v_ref*). The *h_ref/v_ref* selection is controlled via TCON_CTRL1[H_REF_SEL] and TCON_CTRL1[V_REF_SEL]. TCON_CTRL1[VLEN] sets the maximun counter value. The vtgl_counter increments at

the rising edge of *h_ref* when *v_ref* is high, and it falls back to "0" when the counter value exceeds VLEN. And then the vtgl*[0-2]* are generated by decoding the counter value via the following equations.

$$vtgl[0] = (vtgl\_counter == 0)$$

$$vtgl[1] = (vtgl\_counter == 1)$$

$$vtgl[2] = (vtgl\_counter == 2)$$

The H_REF_SEL and V_REF_SEL shall be programmed so that the *h_ref* is a horizontal pulse (one pixel clock cycle pulse per line) and the *v_ref* is a vertical pulse (pulse lasts for 1 line or several lines). The following table shows the relationship between vlen, vtgl_counter sequence and vtgl[2:0].

Table 58-165. vtgl[0-2] sequence

INPUT	OUTPUT	vtgl_counter sequence								
vlen	vtgl	0	1	2	3	4	5	6	7	8
0	0	1	1	1	1	1	1	1	1	1
	1	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	0	1
	1	0	1	0	1	0	1	0	1	0
	2	0	0	0	0	0	0	0	0	0
2	0	1	0	0	1	0	0	1	0	0
	1	0	1	0	0	1	0	0	1	0
	2	0	0	1	0	0	1	0	0	1

A T flip-flop is used to generate vtgl[3]. T flip-flop is a type of flip-flop whose output toggles (inverts) at the rising edge of the clock input when the enable signal input is high. The T flip-flop in the toggle generator is clocked by *h_ref*, and the enable signal (*en*) is the bit-AND of vtgl*[0]* and *v_ref*.

By programming the VLEN, and control the pulse length of v_ref, the vtgl[3] can be used to generate toggle signals whose polarity changes from frame to frame. See the table below for some programming examples.

Table 58-166. vtgl[3] programming examples

vtgl[3]	v_ref pulse length	vlen
toggles every line, polarity inverts frame to frame, steady during vertical blanking (2 frame sequence)	odd (eg. number of active lines + 1)	0
toggles every other line, polarity inverts frame to frame, steady during vertical blanking (2 frame sequence)	odd*2 (eg. number of active lines)	1

Table continues on the next page...

Table 58-166. vtgl[3] programming examples (continued)

vtgl[3]	v_ref pulse length	vlen
toggles every 3rd line, polarity inverts frame to frame, steady during vertical blanking (2 frame sequence)	odd*3 (eg. number of active lines + 3)	2
toggles every other line, walking pattern (4 frame sequence)	odd (eg. number of active lines + 1)	1

To make it easier to understand, the following table shows a simpler example where the number of active lines is 6.

Table 58-167. ctgl[3] signal example A

VLEN	Frame	Active Line Number									v_ref pulse length
		0	1	2	3	4	5	6	7	8	
0	0	0	1	0	1	0	1	0			7
	1	1	0	1	0	1	0	1			7
1	0	0	1	1	0	0	1				6
	1	1	0	0	1	1	0				6
2	0	0	1	1	1	0	0	0	1	1	9
	1	1	0	0	0	1	1	1	0	0	9
1	0	0	1	1	0	0	1	1			7
	1	0	0	1	1	0	0	1			7
	2	1	0	0	1	1	0	0			7
	3	1	1	0	0	1	1	0			7

58.4.2.4 Signal Mixer (SMX)

To generate more complicated timing signals, there are 14 Signal Mixer/Synthesizer (SMX) modules in the TCON which can be used to perform logic operations between every two signals. Each SMX module takes 2 signal inputs (X and Y), and generates 1 SMXx output.

The X and Y input of each SMX module can be selected from any two of the 28 signals (4 comparator outputs, 6 pulses, 4 toggle generator outputs, constant 0 or 1, and 12 feedback signals SMX0~11). The X/Y signal selection is configurable via TCON_SMXx[X_SEL] and TCON_SMXx[Y_SEL]. 12 of the SMXx output (SMX0~11) can be feedback to the SMXx input again in order to generate even more complicated signals (combining information of more than 2 signals),

The SMX module is composed of several logic operation units (AND, OR, XOR or NAND), and a look-up-table (LUT).

The LUT takes 4 inputs, including the current LUT output value and a 4-bit index (index[3:0]), the LUT output is the next cycle SMX output. Function performed by the LUT is determined by the values programmed into TCON_LUTx. By default the value of TCON_LUTx is 32'hffff_005c and the default truth table of the LUT is shown in the following table where SMXx(n-1) is the current SMXx state and SMXx(n) is the next cycle SMXx value. With this setting the LUT emulates the function of a Set/Reset/Toggle/Data flip-flop, where index3 to index0 are the Set/Reset/Toggle enable/Data inputs, "Set" has the highest priority. The user can also reprogram the TCON_LUTx registers so that the LUT performs different operations (e.g. change the priority order of the Set/Reset/Toggle enable/Data signals).

The index[3:0] input of the LUT is generated by the logic operation units. Each bit of it is a logic operation of X, Y, 0 or 1. The logic operations performed are selected by TCON_SMXx[INDEX0-3_SEL].

Table 58-168. Default TCON LUT Truth Table

index3	index2	index1	index0	SMXx(n-1)	SMXx(n)
1	x[1]	x	x	x	1
0	1	x	x	x	0
0	0	1	x	0	1
0	0	1	x	1	0
0	0	0	1	x	1
0	0	0	0	x	0

1. Don't care

58.4.2.5 Output Crossbar Mux

The output crossbar mux is used to select 12 from the generated 20 timing signals as the 12 TCON timing signal outputs. The 20 timing signals include: 6 pulses, 4 toggle generator outputs (vtgl[0:3]), constant "0" and "1", and 8 of the 14 SMXs output (SMX6~13). TCON_OMUX_HIGH and TCON_OMUX_LOW determines which 12 timing signals are selected as the output.

58.4.3 Bit Mapping Control (BMC)

The Bit Mapping Control (BMC) module is adopted to remap the color component order, color bit order and output clock position. The following figure shows the diagram of the bit mapping control module.

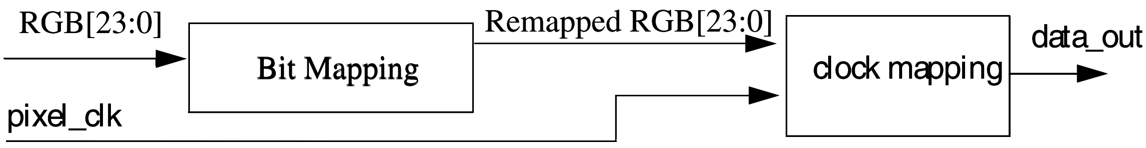


Figure 58-160. Bit mapping control

58.4.3.1 Bit mapping in TTL mode

The default bit mapping in 8-bit color mode is given in the following table.

Table 58-169. TTL mode bit mapping, 8-bit color

data signal	Rising of pix_clk
Remapped RGB[23:16]	{R7,R6,R5,R4,R3,R2,R1,R0}
Remapped RGB[15:8]	{G7,G6,G5,G4,G3,G2,G1,G0}
Remapped RGB[7:0]	{B7,B6,B5,B4,B3,B2,B1,B0}

The default bit mapping in 6-bit colour mode is given in the following table:

Table 58-170. TTL mode bit mapping, 6-bit color

data signal	Rising of pix_clk
Remapped RGB[23:16]	{R7,R6,R5,R4,R3,R2,2'b0}
Remapped RGB[15:8]	{G7,G6,G5,G4,G3,G2,2'b0}
Remapped RGB[7:0]	{B7,B6,B5,B4,B3,B2,2'b0}

Besides the default bit mapping, it is possible to swap the RGB color component order (TCON_BMC[COLOR_ORDER]) and the LSB/MSB bit order in each color component (TCON_BMC[BIT_ORDER]).

58.4.3.2 Bit mapping examples

TTL mode; below settings will remap the bit order as given in the following table.

TCON_BMC[COLOR_ORDER] = 3'b000; //color order RGB

TCON_BMC[BIT_ORDER] = 1; //0 up to MSB7

TCON_CTRL1[COLOR_DEPTH] = 1; //6bit per color;

Table 58-171. Bit mapping example

Data signal	Rising of pix_clk
Remapped RGB[23:16]	{R2,R3,R4,R5,R6,R7,2'b0}
Remapped RGB[15:8]	{G2,G3,G4,G5,G6,G7,2'b0}
Remapped RGB[7:0]	{B2,B3,B4,B5,B6,B7,2'b0}

58.4.3.3 Clock mapping in TTL mode

It is possible to assign the pixel clock output to any bit of the data_out[1:25] in TTL mode, configured via TCON_BMC[CLK_POS]. See the table below for details. The data_out pins are represented on the rows of the table with the number of the pin being in the left hand column. The selected function for that pin is indicated in the row by the CLK_POS value (0 to 24). For example, if CLK_POS = 10 then the function on data_out[11] is clock; if CLK_POS < 10 then data_out[11] is RGB[9]; if CLK_POS > 10 the data_out[11] is RGB[10]

Table 58-172. Clock mapping in TTL mode

<table>
<tr><th rowspan="2">data_out</th><th colspan="25">CLK_POS[4:0]</th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th></tr>
<tr><td>0</td><td colspan="25">const0</td></tr>
<tr><td>1</td><td>C[1]</td><td colspan="24">0</td></tr>
<tr><td>2</td><td>0[2]</td><td>C</td><td colspan="23">1</td></tr>
<tr><td>3</td><td colspan="2">1</td><td>C</td><td colspan="22">2</td></tr>
<tr><td>4</td><td colspan="3">2</td><td>C</td><td colspan="21">3</td></tr>
<tr><td>5</td><td colspan="4">3</td><td>C</td><td colspan="20">4</td></tr>
<tr><td>6</td><td colspan="5">4</td><td>C</td><td colspan="19">5</td></tr>
<tr><td>7</td><td colspan="6">5</td><td>C</td><td colspan="18">6</td></tr>
<tr><td>8</td><td colspan="7">6</td><td>C</td><td colspan="17">7</td></tr>
<tr><td>9</td><td colspan="8">7</td><td>C</td><td colspan="16">8</td></tr>
<tr><td>10</td><td colspan="9">8</td><td>C</td><td colspan="15">9</td></tr>
<tr><td>11</td><td colspan="10">9</td><td>C</td><td colspan="14">10</td></tr>
<tr><td>12</td><td colspan="11">10</td><td>C</td><td colspan="13">11</td></tr>
</table>

Table continues on the next page...

Table 58-172. Clock mapping in TTL mode (continued)

<table>
<tr><th rowspan="2">data_out</th><th colspan="25">CLK_POS[4:0]</th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th></tr>
<tr><td>13</td><td colspan="12">11</td><td>C</td><td colspan="12">12</td></tr>
<tr><td>14</td><td colspan="13">12</td><td>C</td><td colspan="11">13</td></tr>
<tr><td>15</td><td colspan="14">13</td><td>C</td><td colspan="10">14</td></tr>
<tr><td>16</td><td colspan="15">14</td><td>C</td><td colspan="9">15</td></tr>
<tr><td>17</td><td colspan="16">15</td><td>C</td><td colspan="8">16</td></tr>
<tr><td>18</td><td colspan="17">16</td><td>C</td><td colspan="7">17</td></tr>
<tr><td>19</td><td colspan="18">17</td><td>C</td><td colspan="6">18</td></tr>
<tr><td>20</td><td colspan="19">18</td><td>C</td><td colspan="5">19</td></tr>
<tr><td>21</td><td colspan="20">19</td><td>C</td><td colspan="4">20</td></tr>
<tr><td>22</td><td colspan="21">20</td><td>C</td><td colspan="3">21</td></tr>
<tr><td>23</td><td colspan="22">21</td><td>C</td><td colspan="2">22</td></tr>
<tr><td>24</td><td colspan="23">22</td><td>C</td><td>23</td></tr>
<tr><td>25</td><td colspan="24">23</td><td>C</td></tr>
</table>

1. CLOCK
2. n stand for remapped rgb[n]

58.4.4 Clock/Data Skew Adjustment

In order to fulfill different setup/hold timing requirements on the output interface, it is possible to adjust the signal skew between the pixel clock output and the data output (including RGB data and timing signals). The output pixel clock can be flexibly shifted compared to the the data signals. It can be shifted over the whole pixel clock period with a granularity of half of the source pixel clock period (the clock from which the pixel clock is derived/divided).

TCON_CTRL2[DIV_RATIO] configures the pixel clock divide ratio, and TCON_CTRL2[CLK_OFFSET] configures the offset between the internal pixel clock and the output pixel clock.

58.5 Initialization/Application Information

This section discusses TCON initialization and startup.

58.5.1 TCON Initialization

The procedure to bring up the TCON out of reset state and start data and timing signal generation:

1. Program TCON_COMPx and TCON_COMPx_MSK registers to configure the comparator.
2. Program TCON_PULSEx and TCON_PULSEx_MSK registers to configure the pulse generator.
3. Program TCON_SMXx and TCON_LUTx registers if necessary.
4. Program TCON_BMC register to configure the bit mapping control.
5. Program the TCON_CTRL1 register to configure the operation mode of TCON and enable TCON (It is recommended to do this in two steps, and set TCON_EN in the second).

Chapter 59
Run Length Encoding Decoder RLE_DEC

59.1 Introduction

The RLE_DEC module is used to decode data that has been compressed using a Run Length Encoding (RLE) scheme. It has input and output FIFO buffers directly connected to the crossbar switch and requires the CPU or DMA to push in the encoded data and then extract the decoded result. The module configuration is optimized for decoding data stored in a two-dimensional image format but can also be used to extract data stored as a linear array.

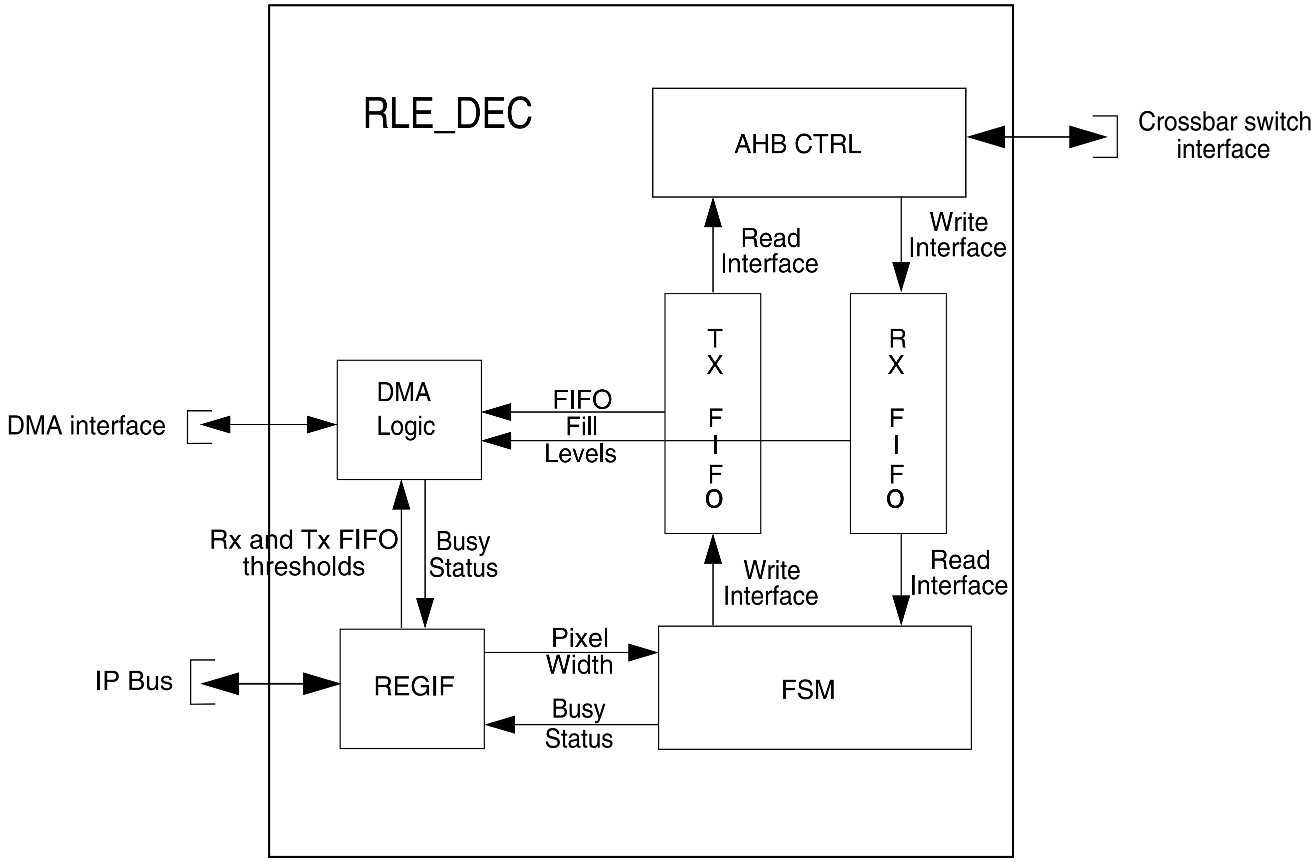


Figure 59-1. RLE_DEC Block Diagram

59.1.1 Overview

The above figure is a block diagram of the Run Length Encoding Decoder (RLE_DEC) module. The module has two independent interfaces that are memory mapped into the device:

- Crossbar switch interface for moving data into and out of the module
- IP bus interface for configuring the module

The crossbar interface appears in the device memory map as two FIFOs. Writes to the Rx FIFO will cause the RLE_DEC to begin decoding operations. The RLE_DECs Finite State Machine (FSM) removes data from the Rx FIFO, decodes it and places the output into the Tx FIFO. The decoded data appears in the Tx FIFO and is removed from this FIFO after it is read by a crossbar master such as the CPU or eDMA. The DMA operations require the use of two different DMA channels.

The module is configured and its status monitored using registers on the IP bus that appear as locations in the module register memory area. This configuration includes information on the size of the image (or data set) and the size of the individual pixels (or data elements) in the image. The RLE_DEC uses this information to determine when an operation is complete. For this reason the RLE_DEC will typically be reconfigured for each operation since the size of each compressed image will be different.

The module can raise interrupts to indicate errors and when an operation is complete.

59.1.2 Features

The RLE_DEC supports the following features:

- Lossless decompression
- Pixel formats supported: 8bpp, 16bpp, 24bpp and 32bpp (Programmable)
- AHB mapped Rx FIFO (8x8 bytes deep) with DMA support.
- AHB mapped Tx FIFO (8x8 bytes deep) with DMA support.
- Programmable fill levels of read and write buffers for initiating burst transfers.
- Partial Image Decode feature, wherein only a portion of the decoded image is given as output. (See id-Image_Co-ordinates_Description for details).
- Support for Stop Mode for power-saving purposes

59.1.3 RLE_DEC Modes of Operation

This section describes the modes of operation.

59.1.3.1 Normal Mode

In this mode, the RLE_DEC block performs the normal 'Run Length Encoding' Decode operation. Further details about this mode of operation can be found in chapter Normal Mode.

59.1.3.2 Module Disable Mode

The Module Disable Mode is used for power management of the device containing the RLE_DEC module. It is controlled by signals external to the RLE_DEC. The clock to the non-memory mapped logic in the RLE_DEC can be stopped while in the Module Disable Mode. See Module Disable Mode.

59.1.3.3 Stop Mode

The Stop Mode is used for power management by the system mode management. When a request is made to enter Stop Mode, the RLE_DEC block completes the action currently processed. Then the request is acknowledged.

59.2 External Signal Description

RLE_DEC has no external signals.

59.3 Interrupt and DMA Request Signals

The interrupt and DMA request lines of the RLE_DEC module are mapped to the internal flags in the DEC_RLE_ISR register.

59.4 Memory map and register definition

The memory map for the RLE_DEC module is given below. The total address for each register is the sum of the base address for the RLE_DEC module and the address offset for each register. Also, please note that the bit order for each register bit field is [highest:lowest], whereas the overall register bits are numbered [lowest:highest].

RLE_DEC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_2000	Module Configuration Register (RLE_DEC_MCR)	32	R/W	2020_0009h	59.4.1/3363
4004_2004	Image Configuration Register (RLE_DEC_ICR)	32	R/W	0000_0000h	59.4.2/3364
4004_2008	Compressed Image Size Register (RLE_DEC_CISR)	32	R/W	0000_0000h	59.4.3/3365
4004_200C	Decompressed Image Co-ordinates register (RLE_DEC_DICR)	32	R/W	0000_0000h	59.4.4/3365

Table continues on the next page...

RLE_DEC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_2010	Status Register (RLE_DEC_SR)	32	R/W	0000_4000h	59.4.5/3366
4004_2014	Interrupt Request Status Register (RLE_DEC_ISR)	32	w1c	0000_0000h	59.4.6/3367
4004_2018	Interrupt Request Enable Register (RLE_DEC_RIER)	32	R/W	0000_0000h	59.4.7/3368
4004_201C	Start Pixel Co-ordinate Register of Image (RLE_DEC_SPCR)	32	R/W	0001_0001h	59.4.8/3369
4004_2020	End Pixel Co-ordinate Register of Image (RLE_DEC_EPCR)	32	R/W	0000_0000h	59.4.9/3369

59.4.1 Module Configuration Register (RLE_DEC_MCR)

The RLE_DEC_MCR holds configuration data associated with RLE_DEC operation. This register can be accessed with 8-bit, 16-bit and 32-bit wide operations.

Address: 4004_2000h base + 0h offset = 4004_2000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	RX_FIFO_THRESHOLD								TX_FIFO_THRESHOLD							
Reset	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved												GRLE_EN	TXFFEN	Reserved	MDIS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

RLE_DEC_MCR field descriptions

Field	Description
31–24 RX_FIFO_ THRESHOLD	Rx FIFO threshold: This field determines how much space should be available for writing into the Rx Buffer until the write action is triggered (through DMA). When the number of bytes of available space in RX FIFO exceeds the number given by this field the RLE_DEC_FR[RFFR] flag is asserted.
23–16 TX_FIFO_ THRESHOLD	Tx FIFO threshold: This field determines how many entries must be read into the TX FIFO until the readout action is triggered. When the number of valid entries in the TX FIFO exceeds the number given by this field the RLE_DEC_FR[TFDR] flag is asserted.
15–4 RESERVED	This field is reserved.
3 GRLE_EN	Enable Gradient RLE decoding This bit enables or disables Gradient RLE decoding. Gradient RLE is described in detail in the section, RLE encoding format. 0 Disables the Gradient RLE decoding. Decoding is performed only as traditional RLE. 1 Enables the Gradient RLE decoding. Decoding can be done according to either Gradient RLE or traditional RLE algorithm, based on the decoding of Command Byte.
2 TXFFEN	Tx FIFO Flush Enable. This bit configures the operation of the RLE Decoder in the case when the image decoding is complete and the data remaining in the Tx FIFO is below the value of TX_FIFO_THRESHOLD. 0 Trigger the DMA action only when TxFIFO has more data than TX_FIFO_THRESHOLD. 1 Trigger the DMA action until the Tx FIFO is empty even if it is less than TX_FIFO_THRESHOLD.
1 RESERVED	This field is reserved.
0 MDIS	Module Disable. The MDIS bit allows the clock to the non-memory mapped logic in the RLE_DEC to be stopped, putting the RLE_DEC in a software controlled power-saving state. See the "Module Disable Mode" section for more information. 0 Enable RLE_DEC clocks. 1 Allow external logic to disable RLE_DEC clocks.

59.4.2 Image Configuration Register (RLE_DEC_ICR)

The RLE_DEC_ICR holds configuration data associated with configuration of image parameters. This register can be accessed with 8-bit, 16-bit and 32-bit wide operations.

Address: 4004_2000h base + 4h offset = 4004_2004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												Reserved		WIDTH	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RLE_DEC_ICR field descriptions

Field	Description
31–4 RESERVED	This field is reserved.
3–2 RESERVED	This field is reserved.
1–0 WIDTH	00 Pixel Width is 08-bits; 01 Pixel Width is 16-bits; 10 Pixel Width is 24-bits; 11 Pixel Width is 32-bits.

59.4.3 Compressed Image Size Register (RLE_DEC_CISR)

The RLE_DEC_CISR holds the size of the compressed image. This register can be accessed with 8-bit, 16-bit and 32-bit wide operations.

Address: 4004_2000h base + 8h offset = 4004_2008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SIZE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RLE_DEC_CISR field descriptions

Field	Description
31–0 SIZE	This is the byte-size of compressed image, that will be given as an input to the RLE DEC. This should be programmed before the Module is enabled using MDIS.

59.4.4 Decompressed Image Co-ordinates register (RLE_DEC_DICR)

The RLE_DEC_DICR holds the final coordinates of the decompressed image. Decompression is done until this pixel reached. Please refer to section "Image Coordinates' Description". This register can be accessed with 8-bit, 16-bit and 32-bit wide operations.

Address: 4004_2000h base + Ch offset = 4004_200Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	X																Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RLE_DEC_DICR field descriptions

Field	Description
31–16 X	This is the X Co-ordinate of the final pixel of the decompressed image, that will be given as the output of the RLE DECODER. This should be programmed before the Module is enabled using MDIS. This field has a minimum value of 1.
15–0 Y	This is the Y Co-ordinate of the final pixel of the decompressed image, that will be given as the output of the RLE DECODER. This should be programmed before the Module is enabled using MDIS. This field has a minimum value of 1.

59.4.5 Status Register (RLE_DEC_SR)

The RLE_DEC_SR register provides the status information about the TX FIFO and RX FIFO.

Address: 4004_2000h base + 10h offset = 4004_2010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																RXFREE								TXFILL							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RLE_DEC_SR field descriptions

Field	Description
31–16 RESERVED	This field is reserved.
15–8 RXFREE	This status gives the amount of free space in bytes, available in Rx FIFO to write.
7–0 TXFILL	This status gives the amount of data in bytes, available in Tx FIFO to read.

59.4.6 Interrupt Request Status Register (RLE_DEC_ISR)

The RLE_DEC_ISR register is the Interrupt Status Register for RLE DECODER. The interrupts are asserted upon associated events. They are deasserted upon write to this register.

Address: 4004_2000h base + 14h offset = 4004_2014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXDIF	TXDIF	RXUIF	TXUIF	RXFIF	TXFIF	RXEIF	TXEIF
W									w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RLE_DEC_ISR field descriptions

Field	Description
31–8 RESERVED	This field is reserved.
7 RXDIF	Rx Image Done. Is asserted when the amount of received data has equalled the programmed "Compressed Image Size Register"
6 TXDIF	Tx Image Done. Is asserted when coordinates of decompressed image cross End Pixel Coordinates which are defined in "End Pixel Coordinate Register"
5 RXUIF	Rx FIFO has space. Asserted when Rx FIFO has more space than Rx FIFO Threshold. (DMA request for input data.)
4 TXUIF	Tx FIFO has data. Asserted when Tx FIFO has more data than Tx FIFO Threshold. (DMA request for output data.)
3 RXFIF	Rx FIFO Full: Asserted when Rx FIFO has no more space left for data and a write was attempted on it.
2 TXFIF	Tx FIFO Full: Asserted when Tx FIFO has no more space left for data and a write was attempted on it.

Table continues on the next page...

RLE_DEC_ISR field descriptions (continued)

Field	Description
1 RXEIF	Rx FIFO Empty: Asserted when Rx FIFO has no data and a read was attempted from it.
0 TXEIF	Tx FIFO Empty: Asserted when Tx FIFO has no data and a read was attempted from it.

59.4.7 Interrupt Request Enable Register (RLE_DEC_RIER)

The RLE_DEC_RIER register provides enables for the interrupts in the RLE_DEC module.

Address: 4004_2000h base + 18h offset = 4004_2018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	Reserved								RXDIE	TXDIE	RXUIE	TXUIE	RXFIE	TXFIE	RXEIE	TXEIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RLE_DEC_RIER field descriptions

Field	Description
31–8 RESERVED	This field is reserved.
7 RXDIE	Interrupt Enable for Rx Image Done Interrupt
6 TXDIE	Interrupt Enable for Tx Image Done Interrupt
5 RXUIE	Interrupt Enable for Rx FIFO has space Interrupt
4 TXUIE	Interrupt Enable for Tx FIFO has data Interrupt
3 RXFIE	Interrupt Enable for Rx FIFO full Interrupt
2 TXFIE	Interrupt Enable for Tx FIFO full Interrupt

Table continues on the next page...

RLE_DEC_RIER field descriptions (continued)

Field	Description
1 RXEIE	Interrupt Enable for Rx FIFO empty Interrupt
0 TXEIE	Interrupt Enable for Tx FIFO empty Interrupt

59.4.8 Start Pixel Co-ordinate Register of Image (RLE_DEC_SPCR)

The RLE_DEC_SPCR holds the start coordinates of the decompressed image. This register can be accessed with 8-bit, 16-bit and 32-bit wide operations.

Address: 4004_2000h base + 1Ch offset = 4004_201Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	X																Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

RLE_DEC_SPCR field descriptions

Field	Description
31–16 X	This is the X Co-ordinate of the first pixel of the decompressed image, that will be given as the output of the RLE DECODER. Please refer to section "Image Coordinates' Description". This should be programmed before the Module is enabled using MDIS. This field has a minimum value of 1.
15–0 Y	This is the Y Co-ordinate of the final pixel of the decompressed image, that will be given as the output of the RLE DECODER. Please refer to section "Image Coordinates' Description". This should be programmed before the Module is enabled using MDIS. This field has a minimum value of 1.

59.4.9 End Pixel Co-ordinate Register of Image (RLE_DEC_EPCR)

The RLE_DEC_SPCR holds the end co-ordinates of the decompressed image of interest. Please refer to section "Image Coordinates' Description". This register can be accessed with 8-bit, 16-bit and 32-bit wide operations.

Address: 4004_2000h base + 20h offset = 4004_2020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	X																Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RLE_DEC_EPCR field descriptions

Field	Description
31–16 X	This is the X Co-ordinate of the last pixel of the decompressed image of interset, that will be given as the output of the RLE DECODER. Please refer to section "Image Coordinates' Description". This should be programmed before the Module is enabled using MDIS. This field has a minimum value of 1.
15–0 Y	This is the Y Co-ordinate of the last pixel of the decompressed image of interset, that will be given as the output of the RLE DECODER. Please refer to section "Image Coordinates' Description". This should be programmed before the Module is enabled using MDIS. This field has a minimum value of 1.

59.4.10 Crossbar Switch Bus Register Memory Map

The table below shows the Rx and Tx FIFOs of the RLE_DEC mapped to the Crossbar switch (AHB) interface.

Table 59-11. RLE_DEC AMBA Bus Memory Map

Address	Register Name
RLE_DEC-AMBA_BASE + 0x0000_0000 to RLE_DEC-AMBA_BASE + 0x0000_0038	Rx FIFO address range:- The AHB master can read from and write into these addresses (For Debug purposes only). The FIFO operation is performed only when the access is made to the below mentioned Memory Mapped Area for Rx FIFO.
RLE_DEC-AMBA_BASE + 0x0000_0040 to RLE_DEC-AMBA_BASE + 0x0000_0078	Tx FIFO address range:- The AHB master can read from and write into these addresses (For Debug purposes only). The FIFO operation is performed only when the access is made to the below mentioned Memory Mapped Area for Tx FIFO.
RLE_DEC-AMBA_BASE + 0x0000_0080 to RLE_DEC-AMBA_BASE + 0x0000_00B8	Memory Mapped Area for Rx FIFO. AHB can only Write into this address. A write into this address results in a write to the Rx FIFO wherever the write pointer points. If Rx FIFO is full, AHB write request will result in HREADY low, unless space is there for the write operation. (Please Note: The write(burst or single write) should start from address (RLE_DEC_AMBA_BASE + 0x0000_0080) only.)
RLE_DEC-AMBA_BASE + 0x0000_00C0 to RLE_DEC-AMBA_BASE + 0x0000_00F8	Memory Mapped Area for Tx FIFO. AHB can only Read from this address. A read into this address results in a read from the top of the Tx FIFO. If Tx FIFO is empty, AHB Read request will result in HREADY low, unless some data is available for reading. (Please Note: The read(burst or single read) should start from address (RLE_DEC_AMBA_BASE + 0x0000_00C0) only.)

59.4.11 Crossbar switch memory map descriptions

This section describes the Rx and Tx FIFO address mapping.

59.4.11.1 Rx FIFO Address Range

The size Rx FIFO is 8-bytes X 8-bytes. The Rx FIFO address range is (RLE_DEC_AMBA_BASE address + 0x0000) to (RLE_DEC_AMBA_BASE address + 0x0038). A crossbar switch can read from and write into these addresses (For Debug purposes only). The FIFO operation is performed only when the access is made to the below mentioned Memory Mapped Area for Rx FIFO.

59.4.11.2 Tx FIFO Address Range

The size Tx FIFO is 8-bytes X 8-bytes. The Tx FIFO address range is (RLE_DEC_AMBA_BASE address + 0x0040) to (RLE_DEC_AMBA_BASE address + 0x0078). A crossbar switch can read from and write into these addresses (For Debug purposes only). The FIFO operation is performed only when the access is made to the below mentioned Memory Mapped Area for Tx FIFO.

59.4.11.3 Memory Mapped Rx FIFO

The contents of Rx FIFO are mapped to (RLE_DEC_AMBA_BASE address + 0x0080). Writing RLE compressed data into the Rx FIFO increments the write-pointer based on the size of the written data.

The RLE decode process can be automated by using the eDMA. Each time the free space of Rx FIFO exceeds a certain level (programmable), a (Rx FIFO) DMA request is asserted. When enabled the eDMA can be used to write RLE compressed data into the Rx FIFO.

A read request to Rx FIFO results in an error.

59.4.11.4 Memory Mapped Tx FIFO

The contents of Tx FIFO are mapped to (RLE_DEC_AMBA_BASE address + 0x00C0). The RLE_DEC module writes decompressed data into the Tx FIFO.

The RLE decode process can be automated by using the eDMA. Each time the contents of the Tx FIFO exceeds a certain fill-level (programmable), a (Tx FIFO) DMA request is asserted. When enabled the eDMA can be used to read the decompressed data from the Tx FIFO.

A write request to Tx FIFO results in an error.

59.5 Functional Description

This section describes the module functions.

59.5.1 RLE encoding format

In the traditional RLE encoding format, the expected encoding is as follows:

- The first data byte in the encoded image is a command byte (CMD[7:0]).
- The ms bit (CMD[7]) indicates if the following bytes are raw or compressed pixels. One pixel can be 8-bit, 16-bit, 24-bit or 32-bit wide, depending on the PIXEL_WIDTH configuration.
- The remaining 7 command bits (CMD[6:0]), specify the "count", i.e. the number of raw or compressed pixels that follow the command byte. The count is offset by 1. Value 0 means count is 1. Value 1 means count is 2, and so on.
- For compressed pixels (CMD[7] = 1), only one pixel follows the command byte. This pixel is repeated count+1 times. A new command follows after CMD+(1*{Pixel width}) pixels.
- For raw pixels (CMD[7] = 0), count+1 number of pixels follow the command byte and these are passed to the Tx FIFO as is.
- If there is more data to decode then a new command follows after CMD+({count +1}*{Pixel width}) bytes. This encoding continues until the whole image is decoded.

In the gradient RLE encoding format, the expected encoding is as follows:

- The first data byte in the encoded image is a command byte (CMD[7:0]).
- For raw pixels (CMD[7] = 0), the encoding is same as traditional RLE.
- For compressed pixels (CMD[7] = 1), the remaining 7 command bits (CMD[6:0]) (or the "count") are used to differentiate between traditional and gradient RLE. If the count is non-zero, the encoding is treated as tradional RLE. If the count is zero, the encoded data is in Gradient RLE format.
- If the encoded data is in Gradient RLE format, the second byte represents the decoded sequence length. If the value of second byte is X, then the decoded sequence length is X+3 pixels. (X+3 is the length of the generated output sequence).

- The pixel that follows represents the first pixel of the decoded sequence. And the pixel after that is the delta. The delta bytes are signed 8-bit fixed point numbers. The second pixel of the decoded sequence is calculated by adding adding delta to the first pixel. The third pixel of the decoded sequence is calculated by adding adding delta to the second pixel, and so on.
- The adding of delta to the pixels is done with help of accumulators. The pixel bytes are loaded into 14-bit accumulators at the MSB. The LSB 6-bits of the accumulatora are 0s. The delta are added to respective accumulators. The MSB 8-bits of the accumulators form the next pixel.

59.5.2 RLE decoding process

The recommended process to decode the data is as follows:

- The RLE_DEC module is disabled by default (MDIS=1).
- Before enabling the device, program the compressed image size register (RLE_DEC_CISR) and decompressed image size register (RLE_DEC_DICR) for the image. The decoder expects to read compressed-image-size amount of bytes from the Rx FIFO and expects to decode and write decompressed-image-size amount of bytes into the Tx FIFO.
- When less than the full image is to be decoded then configure the Start Pixel Coordinates Register (RLE_DEC_SPCR) and the End Pixel Coordinates Register (RLE_DEC_EPCR).
- Configure the WIDTH of the pixels in the image (RLE_DEC_ICR).
- Enable the RLE_DEC DMA Rx and Tx channels in the DMAMUX module and connect these to channels in the eDMA. Configure the selected eDMA channels such that the compressed image is written into the Rx FIFO and the decompressed image is copied from the Tx FIFO.
- Enable the RLE_DEC by setting MDIS=0.
- When the decoding is complete the TXDIF flag is set (RLE_DEC_ISR register). This indicates that the RLE process is complete but depending on the size of the image and the value of the RLE_DEC_MCR[TXFFEN] bit there may be some decoded pixels that have not yet been copied from the Tx FIFO because its threshold has not been reached.

- If the TXFFEN bit is not set to automatically flush the Tx FIFO then trigger an eDMA transfer under software control until the Tx FIFO is empty (TXFILL=0 in RLE_DEC_SR register).
- Set MDIS=1 to disable the module. The module needs to be disabled at the end of image decompression and enabled again only after the parameters of the new image have been programmed.

59.5.3 Image coordinates' example

The following figure shows the definition of the coordinates used in RLE_DEC to define the decompressed image.

Let the decompressed image be of size 10-pixels x 10-pixels. The start coordinate is the coordinate at the top-left corner, (1,1), i.e. X=1 and Y=1. The last coordinate is the bottom-right coordinate, (10,10) in this case. So, we see that the image size can be described by the bottom-right coordinate of the image. This pixel coordinate is described in the register "Decompressed Image Coordinate Register".

If only a part of the complete image is to be given as output, then "Start Pixel Coordinate Register" and "End Pixel Coordinate Register" should be programmed appropriately. For example, in the figure shown below, if the gray area is the image of interest and it is only required to output only that part of the decompressed image, then the "Start Pixel Coordinate Register" should be programmed as (X=3, Y=4) and "End Pixel Coordinate Register" should be programmed as (X=7, Y=7).

If however, the complete decompressed image is to be output, then the "Start Pixel Coordinate Register" should be programmed as (X=1, Y=1) and "End Pixel Coordinate Register" should be programmed as (X=10, Y=10), i.e. equal to "Decompressed Image Coordinate Register".

Table 59-12. Decompressed 10x10 pixel image

(1,1)	(2,1)	(3,1)	(4,1)	(5,1)	(6,1)	(7,1)	(8,1)	(9,1)	(10,1)
(1,2)									
(1,3)									
(1,4)		(3,4)							
(1,5)									
(1,6)									
(1,7)						(7,7)			
(1,8)									
(1,9)									
(1,10)									(10,10)

59.5.4 Modes of Operation

The possible operational modes of the RLE_DEC block are:

- Normal Mode: This is used for normal RLE decompression of data received from AHB interface. The module enters this mode by deasserting RLE_DEC_MCR[MDIS].
- Stop Mode: The mode used by the system when changing modes if the RLE is no longer required. When the system requests Stop Mode, the RLE_DEC block completes the execution of the current command and acknowledges the request. The system then removes clocks to the RLE_DEC block.
- Module Disable Mode: The mode is used for power management. The clock to the non-memory mapped logic in the RLE_DEC can be stopped while in Module Disable Mode. The module enters the mode by setting RLE_DEC_MCR[MDIS].

59.5.5 Normal Mode

In normal mode, decompression of data is performed using RLE scheme. The module enters this mode by deasserting RLE_DEC_MCR[MDIS].

Whenever the Rx FIFO has some data, the state machine is triggered. The decoding operation starts according to the RLE decoding scheme described above. Whenever Tx FIFO is not full and Rx FIFO has got data to transmit, the decoding is done.

After completing the decode of one image, the Normal mode should be exited. For decoding the data of next image, the re-entry to Normal mode should be done only after the new image parameters have been programmed.

59.5.6 Power Saving Features

The RLE_DEC supports the following power-saving strategies:

- Stop Mode
- Module Disable Mode

59.5.6.1 Module Disable Mode

The RLE_DEC block is in Module Disable Mode by default. It is exited to Normal-Mode by de-asserting the MDIS bit in RLE_DEC_MCR register.

Host software can initiate Module Disable Mode by writing a '1' to the MDIS bit. When a request is encountered to enter Module Disable Mode, the RLE_DEC negates clock-enable when it is ready to enter the Module Disable Mode.

If implemented, the clock-enable signal can stop the clock to the non-memory mapped logic. When clock-enable is negated, the RLE_DEC is in a dormant state, but the memory mapped registers are still accessible. Certain read or write operations have a different effect when the RLE_DEC is in the Module Disable Mode. DMA request signals cannot be cleared while in the Module Disable Mode.

Note that issuing a new AHB request is illegal in Normal Mode during the time starting with raising the request to enter Module Disable Mode and ending with leaving the Module Disable Mode.

Chapter 60
Video subsytem

60.1 Introduction

The video subsystem consists of an analog video front end (AFE), a digital video decoder, and video interface unit. The AFE accepts NTSC or PAL input from a device such as an analog camera. The video decoder provides YUV888-formatted data to the the video interface unit (VIU).

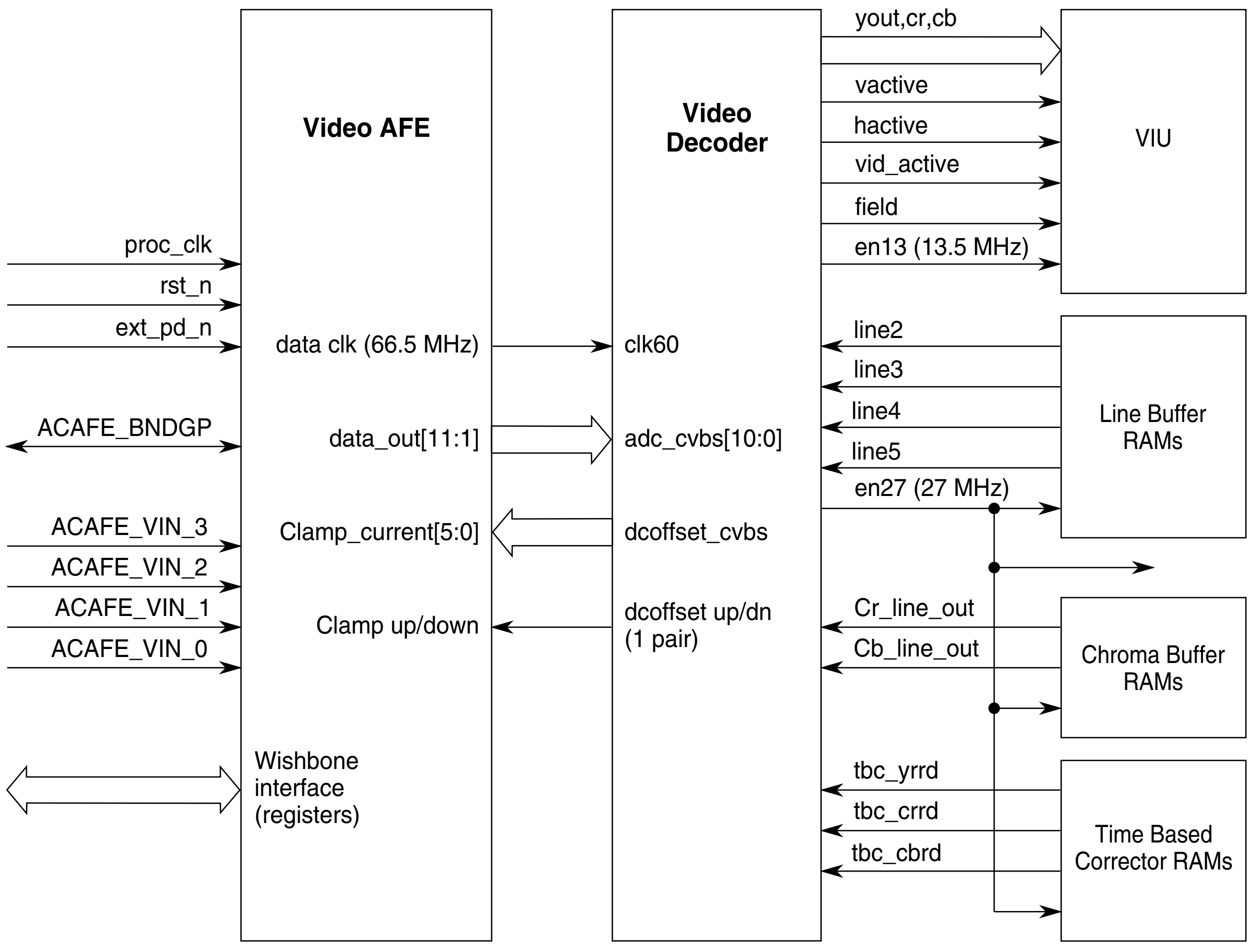


Figure 60-1. Video subsystem block diagram

60.2 External signal description

The following table is a list of signals driven by external pins.

Table 60-1. Video subsystem external signals

Name	Module	Description	Input/ Output
VADCSE0	Video AFE	Composite video input 0	Input
VADCSE1	Video AFE	Composite video input 1	Input
VADCSE2	Video AFE	Composite video input 2	Input
VADCSE3	Video AFE	Composite video input 3	Input
VADC_AFE_BANDGAP	Video AFE	Band gap decoupling	Input/Output

60.3 Analog front end (AFE)

The analog front end (AFE) digitizes an analog video signal such as from an inexpensive analog camera. The video signal can be selected from one of four inputs, VIN0-VIN3, through register control.

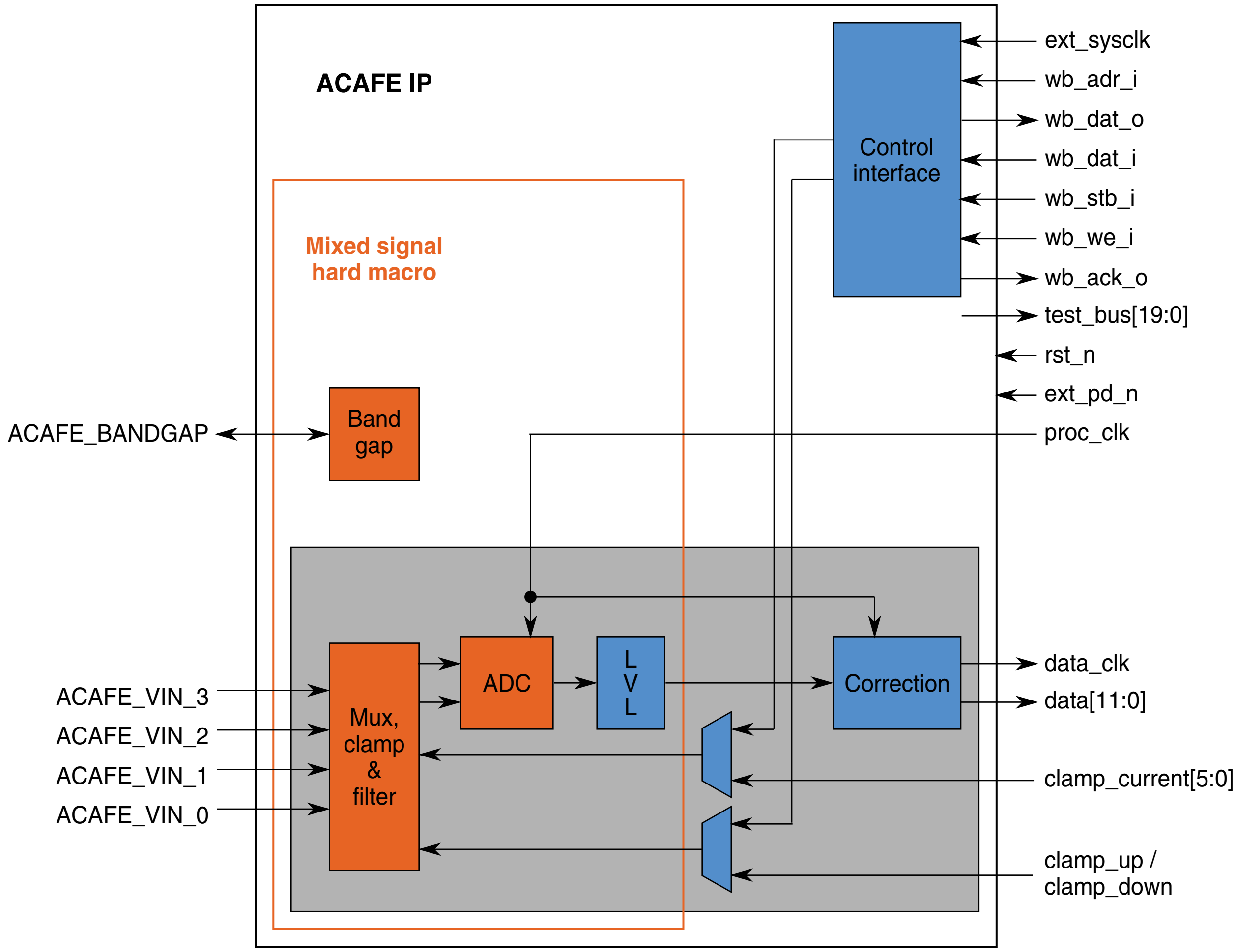


Figure 60-2. Analog front end block diagram

60.3.1 AFE features

The AFE includes the following features.

- Internal voltage and current reference generator
- 10-bit resolution (9.5 bit ENOB at 66.5 Msps)
- 30 mW at 66.5Msps power consumption
- 4 analog inputs. All inputs usable for CVBS.
- Programmable anti-aliasing filter, gain, and clamp

60.4 AFE memory map and registers

NOTE

The AFE registers must be programmed before enabling the Video ADC clock. Refer to the CCM_CSCDR1 register settings in the Clock Control Module chapter for more information.

AFE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_7000	Misc. ID (AFE_MISC_ID)	32	R	0000_0012h	60.4.1/3381
400C_7004	Power Down Buffers (AFE_PDBUF)	32	R/W	0000_0000h	60.4.2/3382
400C_7008	Software Reset (AFE_SWRST)	32	R/W	0000_000Fh	60.4.3/3383
400C_7018	Band Gap (AFE_BGREG)	32	R/W	0000_0008h	60.4.4/3384
400C_7400	Accessar ID (AFE_ACCESSAR_ID)	32	R	0000_0011h	60.4.5/3384
400C_7404	Power Down ADC (AFE_PDADC)	32	R/W	0000_0000h	60.4.6/3385
400C_7408	Power Down SAR High (AFE_PDSARH)	32	R/W	0000_0000h	60.4.7/3386
400C_740C	Power Down SAR Low (AFE_PDSARL)	32	R/W	0000_0000h	60.4.8/3386
400C_7410	Power Down ADC Ref. High (AFE_PDADCRFH)	32	R/W	0000_0000h	60.4.9/3387
400C_7414	Power Down ADC Ref. Low (AFE_PDADCRFL)	32	R/W	0000_0000h	60.4.10/ 3387
400C_741C	ADC Gain (AFE_ADCGN)	32	R/W	0000_000Fh	60.4.11/ 3388
400C_7434	ADC Ref Trim Low (AFE_REFTRIML)	32	R/W	0000_0055h	60.4.12/ 3388
400C_7438	ADC Ref Trim High (AFE_REFTRIMH)	32	R/W	0000_0000h	60.4.13/ 3389
400C_7448	Delay Loop Calculated Data (AFE_DLYALG)	32	R	0000_0000h	60.4.14/ 3390
400C_744C	Clamp DAC Trim (AFE_DACAMP)	32	R/W	0000_0000h	60.4.15/ 3390
400C_7454	Clamp DAC Data (AFE_CLMPDAT)	32	R/W	0000_0000h	60.4.16/ 3391
400C_7458	Clamp DAC Control (AFE_CLMPAMP)	32	R/W	0000_0000h	60.4.17/ 3392
400C_745C	Clamp Control (AFE_CLAMP)	32	R/W	0000_0000h	60.4.18/ 3393
400C_7460	Input Buffer (AFE_INPBUF)	32	R/W	0000_0000h	60.4.19/ 3394

Table continues on the next page...

AFE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_7464	Analog Input Filter (AFE_INPFLT)	32	R/W	0000_0000h	60.4.20/ 3395
400C_7468	ADC Digital Gain (AFE_ADCDGN)	32	R/W	0000_0000h	60.4.21/ 3397
400C_746C	Off-Chip Drive (AFE_OFFDRV)	32	R/W	0000_0000h	60.4.22/ 3397
400C_7800	Acc ID (AFE_ACC_ID)	32	R	0001_0011h	60.4.23/ 3398
400C_7808	ADC Sample Acquisition (AFE_ASAREG)	32	R/W	0000_0000h	60.4.24/ 3399
400C_7810	ADC Sample Compensation (AFE_ASCREG)	32	R/W	0000_0000h	60.4.25/ 3400
400C_7814	Block Level Control Register (AFE_BLCREG)	32	R/W	0000_0000h	60.4.26/ 3401
400C_7824	ADC Operation Controller 0 (AFE_AOCREG0)	32	R/W	0000_0000h	60.4.27/ 3402

60.4.1 Misc. ID (AFE_MISC_ID)

Address: 400C_7000h base + 0h offset = 400C_7000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								misc_id							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

AFE_MISC_ID field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 misc_id	Misc. ID and revision number test

60.4.2 Power Down Buffers (AFE_PDBUF)

Address: 400C_7000h base + 4h offset = 400C_7004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											Reserved		bgr_pd_n	bgr_bgr_pd_n	acafe_pd_n
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_PDBUF field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–3 Reserved	This field must remain set to reset value, 00b. This field is reserved.
2 bgr_pd_n	Active-low power down of band gap. Both bgr_pd_n and bgr_bgr_pd_n must be enabled for the bandgap to be in power up. 0 Power down 1 Normal
1 bgr_bgr_pd_n	Active-low power down of band gap core. Both bgr_pd_n and bgr_bgr_pd_n must be enabled for the bandgap to be in power up. 0 Power down 1 Normal
0 acafe_pd_n	Active-low power down of IP 0 Power down 1 Normal

60.4.3 Software Reset (AFE_SWRST)

Address: 400C_7000h base + 8h offset = 400C_7008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												acafe_sw_rst_n	adc_proc_clk_sw_rst_n	Reserved	sysclk_sw_rst_n
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

AFE_SWRST field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 acafe_sw_rst_n	Software reset of all clocks 0 Reset 1 Normal
2 adc_proc_clk_sw_rst_n	Software reset of adc_clk 0 Reset 1 Normal
1 Reserved	This field must remain set to 1. This field is reserved.
0 sysclk_sw_rst_n	Software reset of sysclk 0 Reset 1 Normal

60.4.4 Band Gap (AFE_BGREG)

Address: 400C_7000h base + 18h offset = 400C_7018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											Reserved	bgr_trimlevel			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

AFE_BGREG field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 Reserved	This field must remain set to 0. This field is reserved.
3–0 bgr_trimlevel	Trim of bandgap Value ranges from 0 (minimum) to 15 (maximum). **NOTE:** After a power-on reset, functional reset, or standby exit, this field should be written with the value in the OCOTP Controller register OCOTP_RNG[VADC_BANDGAP] after the OCOTP_CTRL[BUSY] has cleared to 0.

60.4.5 Accessar ID (AFE_ACCESSAR_ID)

Address: 400C_7000h base + 400h offset = 400C_7400h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								accessar_id							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

AFE_ACCESSAR_ID field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 accessar_id	Accessar ID

60.4.6 Power Down ADC (AFE_PDADC)

Address: 400C_7000h base + 404h offset = 400C_7404h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											clamp_pd_n	adc_iref_pd_n	adc_dlyloop_dac_pd_n	dlyloop_pd_n	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_PDADC field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 clamp_pd_n	Active-low power down of clamp circuitry 0 Power down 1 Normal
3 adc_iref_pd_n	Active-low power down of ADC iref 0 Power down 1 Normal
2 adc_dlyloop_dac_pd_n	Active-low power down of ADC delay loop DAC 0 Power down 1 Normal

Table continues on the next page...

AFE_PDADC field descriptions (continued)

Field	Description
1 dlyloop_pd_n	Active-low power down of ADC delay loop reference 0 Power down 1 Normal
0 Reserved	This field is reserved.

60.4.7 Power Down SAR High (AFE_PDSARH)

Address: 400C_7000h base + 408h offset = 400C_7408h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															adc_pd_n
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_PDSARH field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 adc_pd_n	Active-low power down of ADC, one bit per SAR ADC 0 Power down 1 Normal

60.4.8 Power Down SAR Low (AFE_PDSARL)

Address: 400C_7000h base + 40Ch offset = 400C_740Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								adc_pd_n							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_PDSARL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

AFE_PDSARL field descriptions (continued)

Field	Description
7–0 adc_pd_n	Active-low power down of ADC, one bit per SAR ADC 0 Power down 1 Normal

60.4.9 Power Down ADC Ref. High (AFE_PDADCRFH)

Address: 400C_7000h base + 410h offset = 400C_7410h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															adcref_refbufslice_pd_n
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_PDADCRFH field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 adcref_ refbufslice_pd_n	Active-low power down of ADC reference, one bit per SAR ADC 0 Power down 1 Normal

60.4.10 Power Down ADC Ref. Low (AFE_PDADCRFL)

Address: 400C_7000h base + 414h offset = 400C_7414h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								adcref_refbufslice_pd_n							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_PDADCRFL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 adcref_ refbufslice_pd_n	Active-low power down of ADC reference, one bit per SAR ADC 0 Power down 1 Normal

60.4.11 ADC Gain (AFE_ADCGN)

Address: 400C_7000h base + 41Ch offset = 400C_741Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												adc_gain			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

AFE_ADCGN field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 adc_gain	ADC gain setting Value of zero equals no signal. Value of one equals gain of one. Value of 15 equals gain of 15.

60.4.12 ADC Ref Trim Low (AFE_REFTRIML)

Address: 400C_7000h base + 434h offset = 400C_7434h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								adcref_ reftrimop		adcref_ reftrim02		adcref_ reftrim04		adcref_ reftrim08	
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

AFE_REFTRIML field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

AFE_REFTRIML field descriptions (continued)

Field	Description
7–6 adcref_reftrimop	ADC reference Trim reference buffers multiplier 00 X1 01 X2 10 X2 11 X3
5–4 adcref_reftrim02	ADC reference Trim 0.2V reference 00 Low 01 Mid 10 Mid 11 High
3–2 adcref_reftrim04	ADC reference Trim 0.8V common-mode reference 00 Low 01 Mid 10 Mid 11 High
1–0 adcref_reftrim08	ADC reference Trim 0.8V reference 00 Low 01 Mid 10 Mid 11 High

60.4.13 ADC Ref Trim High (AFE_REFTRIMH)

Address: 400C_7000h base + 438h offset = 400C_7438h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												adcref_reftrim			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_REFTRIMH field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

AFE_REFTRIMH field descriptions (continued)

Field	Description
3–0 adcref_reftrim	ADC reference Binary coded master trim. Vref_master(x) = Vref * (12+0.25*x)/14 0 X0.86 1 X0.88 2 X0.89 3 X0.91 4 X0.93 5 X0.95 6 X0.96 7 X0.98 8 X1.00 9 X1.02 10 X1.04 11 X1.05 12 X1.07 13 X1.09 14 X1.11 15 X1.13

60.4.14 Delay Loop Calculated Data (AFE_DLYALG)

Address: 400C_7000h base + 448h offset = 400C_7448h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									dlyloop_calculateddata						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_DLYALG field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–0 dlyloop_ calculateddata	Tuning ADC timing algorithm. Value found by algorithm.

60.4.15 Clamp DAC Trim (AFE_DACAMP)

Address: 400C_7000h base + 44Ch offset = 400C_744Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												clampdac_ trim			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_DACAMP field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 clampdac_trim	Trim of DAC current Current is proportional to this value. Zero is lowest current. 15 is highest current.

60.4.16 Clamp DAC Data (AFE_CLMPDAT)

Address: 400C_7000h base + 454h offset = 400C_7454h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								clampdac_data							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_CLMPDAT field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 clampdac_data	Clamp DAC data Current is proportional to this value. 0 is lowest current, 255 is highest current. Only valid when clamp_current_reg_override is set.

60.4.17 Clamp DAC Control (AFE_CLMPAMP)

Address: 400C_7000h base + 458h offset = 400C_7458h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									clamp_current_reg_override	clamp_updn_reg_override	clamp_dacdata_weight		clamp_dacdata_extra		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_CLMPAMP field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 clamp_current_reg_override	Override clamp current ports and control through registers. 0 Ports 1 Register
5 clamp_updn_reg_override	Override clamp up down ports and control through registers. 0 Ports 1 Register
4–3 clamp_dacdata_weight	Maps clamp_current input to clampdac_data port in mixed-signal block. 00 No shift 01 Shift by 1 10 Shift by 2 11 Shift by 3
2–0 clamp_dacdata_extra	Clamp DAC extra data Defines non-assigned bits when in current leakage mode. Fills out missing bits when mapping from five bits to eight bits.

60.4.18 Clamp Control (AFE_CLAMP)

Address: 400C_7000h base + 45Ch offset = 400C_745Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								clamp_pwm_mode	clamp_up_down_polarity	clamp_irefselect	clamp_lowcurrmode	clamp_inen_reg	clamp_ipen_reg	Reserved	nclamp_powersave
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_CLAMP field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 clamp_pwm_mode	Enable PWM mode 0 Constant 1 PWM
6 clamp_up_down_polarity	Defines polarity of MSB in clamp_current port. 0 Non-inverted 1 Inverted
5 clamp_irefselect	Enable current reference to the clamp DAC. 0 Off 1 On
4 clamp_lowcurrmode	Enable low current mode 0 Normal 1 Low current
3 clamp_inen_reg	Clamp down. Remove charge. Only valid when clamp_updn_reg_override. 0 No pump down 1 Pump down
2 clamp_ipen_reg	Clamp up. Add charge. Only valid when clamp_updn_reg_override.

Table continues on the next page...

AFE_CLAMP field descriptions (continued)

Field	Description
	0 No pump up 1 Pump up
1 Reserved	This field must remain set to 0. This field is reserved.
0 nclamp_ powersave	Active-low power save 0 Power save 1 Normal

60.4.19 Input Buffer (AFE_INPBUF)

Address: 400C_7000h base + 460h offset = 400C_7460h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										mux_clampen	mux_buffer_15m_en	mux_buffer_bp_en	buff_en_cm	Reserved	buff_en_ri
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_INPBUF field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 mux_clampen	Connect clamp node to analog input. 0 Disable 1 Enable
4 mux_buffer_ 15m_en	15MHz buffer enable Selects 15 MHz low-pass filter when set together with INPFLT[mux_filter_15m_en]. 0 Disable filter 1 Enable filter

Table continues on the next page...

AFE_INPBUF field descriptions (continued)

Field	Description
3 mux_buffer_bp_en	Buffer bypass enable Bypasses filter when this field and AFE_INPFLT[mux_filterbypass] are set to 1. 0 Disable 1 Enable
2 buff_en_cm	Common mode input buffer enable Enable common-mode output of analog input buffer. 0 Disable 1 Enable
1 Reserved	This field must remain set to 0. This field is reserved.
0 buff_en_ri	Differential output buffer enable Enable differential output of analog input buffer. 0 Disable 1 Enable

60.4.20 Analog Input Filter (AFE_INPFLT)

Address: 400C_7000h base + 464h offset = 400C_7464h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									mux_enlf				mux_filterbypass	mux_filter_15m_en	mux_pdcurrentmirror
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_INPFLT field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–3 mux_enlf	Analog input enable Enables (when one) or disables (when zero) analog input ports. Each port is mapped to its respective bit in this field. Bit 3 Port 0 Bit 4 Port 1 Bit 5 Port 2 Bit 6 Port 3 For example, to enable ports 0 and 2 and disable ports 1 and 3, write 0x5 to this field.
2 mux_filterbypass	Fiter bypass Bypasses filter when set together with AFE_INPBUF[mux_buffer_bp_en]. 0 Disable bypass 1 Enable bypass
1 mux_filter_15m_en	15 MHz filter enable Selects 15 MHz low-pass filter when set together with AFE_INPBUF[mux_buffer_15m_en]. 0 Disable filter 1 Enable filter
0 mux_ pdcurrentmirror	Power down current mirror Enable buffter current mirrors 0 Power down 1 Normal

60.4.21 ADC Digital Gain (AFE_ADCDGN)

Address: 400C_7000h base + 468h offset = 400C_7468h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									adc_digital_gain_bypass	adc_digital_gain					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_ADCDGN field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 adc_digital_gain_bypass	Bypass digital gain 0 Normal 1 Bypass
5–0 adc_digital_gain	ADC digital gain Gain is equal to 1 + adc_digital_gain/256.

60.4.22 Off-Chip Drive (AFE_OFFDRV)

Address: 400C_7000h base + 46Ch offset = 400C_746Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								input_pulldown_en				Reserved			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_OFFDRV field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 input_pulldown_en	Input pull-down enable Enable (1) or disable (0) pull down input ports. Each port is mapped to its respective bit and should be set to the inverse of AFE_INPLFT[mux_enlf]. Bit 4 Pull-down input 0 Bit 5 Pull-down input 1 Bit 6 Pull-down input 2 Bit 7 Pull-down input 3
3–0 Reserved	This field is reserved.

60.4.23 Acc ID (AFE_ACC_ID)

Address: 400C_7000h base + 800h offset = 400C_7800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								acc_id							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

AFE_ACC_ID field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 acc_id	Block ID and revision number

60.4.24 ADC Sample Acquisition (AFE_ASAREG)

Address: 400C_7000h base + 808h offset = 400C_7808h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								asa_reg			Reserved	asa_reg9	asa_reg5	asa_reg3	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_ASAREG field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 asa_reg	ADC sample acquisition
4 Reserved	This field is reserved.
3 asa_reg9	ADC sample acquisition. Enable 9 slices. Only one of bits 4 down-to 1 can be set. 0 Disable 1 Enable 9 Slices
2 asa_reg5	ADC sample acquisition. Enable 5 slices. Only one of bits 4 down-to 1 can be set. 0 Disable 1 Enable 5 Slices
1 asa_reg3	ADC sample acquisition. Enable 3 slices. Only one of bits 4 down-to 1 can be set. 0 Disable 1 Enable 3 slices
0 Reserved	This field is reserved.

60.4.25 ADC Sample Compensation (AFE_ASCREG)

Address: 400C_7000h base + 810h offset = 400C_7810h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								Reserved							asc_reg
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_ASCREG field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–1 Reserved	This field is reserved.
0 asc_reg	ADC offset compensation Disables digital slice-to-slice offset compensation. 0 Enable slice-to-slice offset compensation 1 Disable slice-to-slice offset compensation

60.4.26 Block Level Control Register (AFE_BLCREG)

Address: 400C_7000h base + 814h offset = 400C_7814h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								Reserved						start_cal	en_bypass
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_BLCREG field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–2 Reserved	This field is reserved.
1 start_cal	Start calibration Starts calibration when set to 1. 0 Normal 1 Start
0 en_bypass	Enable bypass 0 Normal 1 Bypass

60.4.27 ADC Operation Controller 0 (AFE_AOCREG0)

Address: 400C_7000h base + 824h offset = 400C_7824h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								Reserved							start_cal
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AFE_AOCREG0 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–1 Reserved	This field is reserved.
0 start_cal	Start calibration 0 Normal 1 Start

60.5 Video decoder

The video decoder is comprised of the following blocks.

- DVB/ATSC demodulator
- NTSC/PAL IF demodulator
- NTSC/PAL decoder
- Video processing, noise reduction, de-interlacing, and scaling

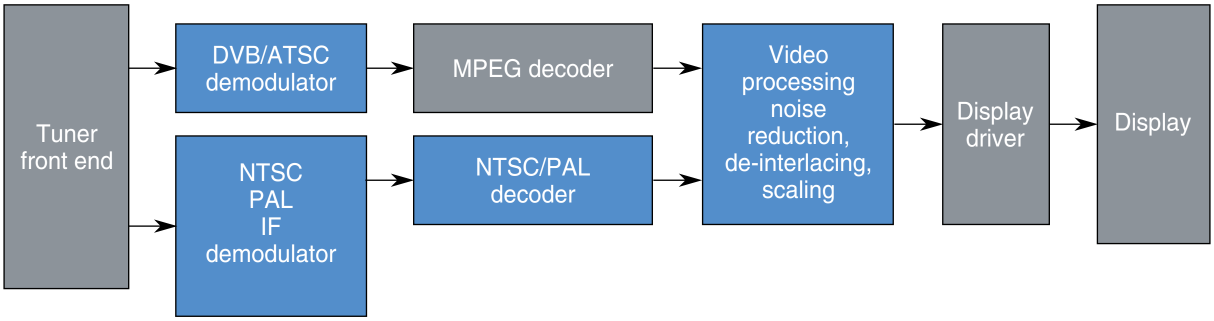


Figure 60-30. Video decoder block diagram

60.5.1 Video decoder features

The video decoder includes the following features.

- NTSC/PAL decoder
- Direct data path (no complex resampling)
- Automatic standards detection
- 2D adaptive comb filter
- Datapath/clocking architecture encompasses a time base corrector for VCR signals
- Luma passband is flat to >6MHz

60.6 Video decoder memory map and registers

VDEC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_8000	2D Comb Filter Control 1 (VDEC_CFC1)	32	R/W	0000_0000h	60.6.1/3405
400C_8024	Burst Gate (VDEC_BRSTGT)	32	R/W	0000_0000h	60.6.2/3406
400C_8040	Horizontal Position (VDEC_HZPOS)	32	R/W	0000_0000h	60.6.3/3406
400C_8044	Vertical Position (VDEC_VRTPOS)	32	R/W	0000_0000h	60.6.4/3407
400C_8054	Output Conditioning and HV Shift (VDEC_HVSHFT)	32	R/W	0000_0000h	60.6.5/3407
400C_8058	HSync Ignore Start (VDEC_HSIGS)	32	R/W	0000_0000h	60.6.6/3408
400C_805C	HSync Ignore End (VDEC_HSIGE)	32	R/W	0000_0000h	60.6.7/3409
400C_8060	VSync Control 1 (VDEC_VSCON1)	32	R/W	0000_0000h	60.6.8/3409
400C_8064	VSync Control 2 (VDEC_VSCON2)	32	R/W	0000_0000h	60.6.9/3411
400C_806C	Y/C Delay and Chroma Debug (VDEC_YCDEL)	32	R/W	0000_0000h	60.6.10/ 3412
400C_8070	After Clamp (VDEC_AFTCLP)	32	R/W	0000_0000h	60.6.11/ 3413

Table continues on the next page...

VDEC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_8078	DC Offset (VDEC_DCOFF)	32	R/W	0000_0000h	60.6.12/ 3414
400C_8084	Chroma Swap, Invert, and Debug (VDEC_CSID)	32	R/W	0000_0000h	60.6.13/ 3415
400C_8088	Cb Gain (VDEC_CBGN)	32	R/W	0000_0000h	60.6.14/ 3416
400C_808C	Cr Gain (VDEC_CRGN)	32	R/W	0000_0000h	60.6.15/ 3416
400C_8090	Contrast (VDEC_CNTR)	32	R/W	0000_0000h	60.6.16/ 3416
400C_8094	Brightness (VDEC_BRT)	32	R/W	0000_0000h	60.6.17/ 3417
400C_8098	Hue (VDEC_HUE)	32	R/W	0000_0000h	60.6.18/ 3417
400C_809C	Chroma Burst Threshold (VDEC_CHBTH)	32	R/W	0000_0000h	60.6.19/ 3418
400C_80A4	Sharpness Improvement (VDEC_SHPIMP)	32	R/W	0000_0000h	60.6.20/ 3418
400C_80A8	Chroma PLL and Input Mode (VDEC_CHPLLIM)	32	R/W	0000_0000h	60.6.21/ 3419
400C_80AC	Video Mode (VDEC_VIDMOD)	32	R	Undefined	60.6.22/ 3420
400C_80B0	Video Status (VDEC_VIDSTS)	32	R	0000_0000h	60.6.23/ 3422
400C_80B4	Noise Detector (VDEC_NOISE)	32	R	0000_0000h	60.6.24/ 3423
400C_80B8	Standards and Debug (VDEC_STDDBG)	32	R/W	0000_0000h	60.6.25/ 3423
400C_80BC	Manual Override (VDEC_MANOVR)	32	R/W	0000_0000h	60.6.26/ 3425
400C_80C8	VSync and Signal Thresholds (VDEC_VSSGTH)	32	R/W	0000_0000h	60.6.27/ 3426
400C_80D0	Debug Framebuffer (VDEC_DBGFBH)	32	R/W	0000_0000h	60.6.28/ 3427
400C_80D4	Debug Framebuffer 2 (VDEC_DBGFBL)	32	R/W	0000_0000h	60.6.29/ 3427
400C_80D8	H Active Start (VDEC_HACTS)	32	R/W	0000_0000h	60.6.30/ 3428
400C_80DC	H Active End (VDEC_HACTE)	32	R/W	0000_0000h	60.6.31/ 3428
400C_80E0	V Active Start (VDEC_VACTS)	32	R/W	0000_0000h	60.6.32/ 3428
400C_80E4	V Active End (VDEC_VACTE)	32	R/W	0000_0000h	60.6.33/ 3429

Table continues on the next page...

VDEC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400C_80EC	HSync Tip (VDEC_HSTIP)	32	R/W	0000_0000h	60.6.34/ 3429
400C_80F8	Bluescreen Cr (VDEC_BLSCRCR)	32	R/W	0000_0000h	60.6.35/ 3430
400C_80FC	Bluescreen Cb (VDEC_BLSCRCB)	32	R/W	0000_0000h	60.6.36/ 3430
400C_8104	Luma AGC Control 2 (VDEC_LMAGC2)	32	R/W	0000_0000h	60.6.37/ 3430
400C_810C	Chroma AGC Control 2 (VDEC_CHAGC2)	32	R/W	0000_0000h	60.6.38/ 3431
400C_8114	Minimum Threshold (VDEC_MINTH)	32	R/W	0000_0000h	60.6.39/ 3431
400C_811C	Vertical Lines High (VDEC_VFRQOH)	32	R	0000_0000h	60.6.40/ 3432
400C_8120	Vertical Lines Low (VDEC_VFRQOL)	32	R	Undefined	60.6.41/ 3432

60.6.1 2D Comb Filter Control 1 (VDEC_CFC1)

Address: 400C_8000h base + 0h offset = 400C_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								rc_debugout				rc_combmode_override			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_CFC1 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 rc_debugout	Debug mode 000 Full 2D comb filter 001 Vertical adaptive comb only 010 Fixed 3 line vertical comb only 011 Fixed notch filter only 100 Reserved 101 Reserved

Table continues on the next page...

VDEC_CFC1 field descriptions (continued)

Field	Description
	110 Reserved 111 Reserved
3–0 rc_combmode_ override	Comb mode override Overrides the automatic comb mode for various standards. Bit – Description 0 – Enable override 1 – Reserved 2 – Force 4.43 MHz comb filters 3 – Force PAL 2D comb mode

60.6.2 Burst Gate (VDEC_BRSTGT)

Address: 400C_8000h base + 24h offset = 400C_8024h

Bit	31–8	7–0
R	0	rc_cburststart
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

VDEC_BRSTGT field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rc_cburststart	Burst start position This sets the starting position of the burst measurement gate.

60.6.3 Horizontal Position (VDEC_HZPOS)

Address: 400C_8000h base + 40h offset = 400C_8040h

Bit	31–8	7–0
R	0	ro_hpramp_cmp
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

VDEC_HZPOS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 ro_hpramp_cmp	Horizontal position Horizontal position of output. Signed number in pixels.

60.6.4 Vertical Position (VDEC_VRTPOS)

Vertical position of output.

Address: 400C_8000h base + 44h offset = 400C_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								ro_vline_cmp							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_VRTPOS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 ro_vline_cmp	Vertical position Vertical position of output. Signed number in lines.

60.6.5 Output Conditioning and HV Shift (VDEC_HVSHFT)

Address: 400C_8000h base + 54h offset = 400C_8054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									antialias_dis	ro_useactive	ro_vzero_sel	Reserved		ro_invfield	ro_hzero_sel
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_HVSHFT field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 antialias_dis	Anti-alias disable When set to 1, anti-alias filter is bypassed.
5 ro_useactive	Use active Use the output active video signals to enable and blank the output video.
4 ro_vzero_sel	Vertical shift When set to 1, the output screen is shifted vertically so the V blank areas can be set for debug.
3–2 Reserved	This field is reserved.
1 ro_invfield	Invert field Invert the field output pin.
0 ro_hzero_sel	Horizontal shift When set to 1, the output screen will be shifted by half a screen horizontally so the H blank areas can be set for debug.

60.6.6 HSync Ignore Start (VDEC_HSIGS)

Address: 400C_8000h base + 58h offset = 400C_8058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rv_ignorestart							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_HSIGS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rv_ignorestart	Ignore start Line number (in half lines) of the ignore period. During this period, the Hysncs and DC offset are not monitored.

60.6.7 HSync Ignore End (VDEC_HSIGE)

Address: 400C_8000h base + 5Ch offset = 400C_805Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rv_ignoreend							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_HSIGE field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rv_ignoreend	Ignore end Line number (in half lines) of the ignore period. During this period, the Hysncs and DC offset are not monitored.

60.6.8 VSync Control 1 (VDEC_VSCON1)

Address: 400C_8000h base + 60h offset = 400C_8060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								rh_8or16	rh_modadd_dis	rh_vsynchalfmode	rh_dis_vsyncdetect	rh_robust625det	rh_vdet_dbg		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_VSCON1 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 rh_8or16	Vsync detector

Table continues on the next page...

VDEC_VSCON1 field descriptions (continued)

Field	Description
	Select vsync detector 0 new 16 long vsync detector 1 old shorter vsync detector
6 rh_modadd_dis	Debug Debug test mode only
5 rh_ vsynchalfmode	Vsync half mode Use half lines instead of quarter lines.
4 rh_dis_ vsyncdetect	Disable vsync detection When set, do not look for any new vsync phases. Coast in previously found phase.
3 rh_robust625det	Robust 625 detection Selects a more robust method of determining 525 or 625 line mode. In particular this will coast in the last known mode in the event of no signal.
2–0 rh_vdet_dbg	Vsync debug mode Sets the Vsync debug output mode that can be seen by setting VIDOUTDBG[r_sel2] to 1. 0 Predicted location of Vsync 1 High when a new Vphase is set in the IP 2 Very simple Vsync detector for debug output only 3 High when several valid Vsyncs in a row are detected 4 High when this field's possible Vsync phases are looked at 5 The many detected Vsyncs (false and real) 6 High when twice as many Vsync phases are matching (i.e., twice as many as the 3 debug output). 7 Low-pass filtered luma falling edge detector. This is a crude Vsync detector used in conjunction with other methods to remove false detections

60.6.9 VSync Control 2 (VDEC_VSCON2)

Address: 400C_8000h base + 64h offset = 400C_8064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								rh_disable_hsw	rh_smooth_hsw	rh_hsw_ coring		rh_vcr_ force_dis		rh_vcr_ phasethr	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_VSCON2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 rh_disable_hsw	Head switch detection disable Disable detection of VCR head switches 0 New 16 long Vsync detector 1 Old, shorter Vsync detector
6 rh_smooth_hsw	Headswitch smoothing 0 Use the phase of one new line after a headswitch 1 Use the average of 4 lines for the new phase after a headswitch
5–4 rh_hsw_coring	Headswitch coring value Coring value for phase variance measurement in headswitch detector. This prevents detectioning headswitchs on noise.
3–2 rh_vcr_force_dis	Override VCR detect mode 00 Automatic detection 01 Disable VCR detection 10 Force VCR mode 11 Reserved
1–0 rh_vcr_phasethr	VCR detection threshold The higher the value the less likely that we will detect a VCR (ie the more erratic a VCR needs to be to be detected).

60.6.10 Y/C Delay and Chroma Debug (VDEC_YCDEL)

Address: 400C_8000h base + 6Ch offset = 400C_806Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								rd_lumadel				Reserved	rd_wide	rd_narrow	rd_nopalave
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_YCDEL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 rd_lumadel	Luma delay Luma delay relative to chroma in half pixel increments 0 new 16 long vsync detector 1 old shorter vsync detector
3 Reserved	This field is reserved.
2 rd_wide	Wide mode Force the chroma output low pass filter into wide mode. Undefined if this field and rd_narrow are both set to 1.
1 rd_narrow	Narrow mode Force the chroma output low-pass filter into narrow mode. Undefined if this field and rd_wide are both set to 1.
0 rd_nopalave	No PAL averaging Turn off the two-line PAL chroma averaging. For debug only.

60.6.11 After Clamp (VDEC_AFTCLP)

Address: 400C_8000h base + 70h offset = 400C_8070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									rc_aoutoafterclamp_dis	rc_midfield_dis	rc_afterclamp_update_en	0	rl_resetoffset	rl_disoffset	rh_shortframe
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_AFTCLP field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 rc_aoutoafterclamp_dis	Auto after clamp disable Disables the auto-level measurement in the after-clamp block.
5 rc_midfield_dis	Midfield update disable When rc_afterclamp_update_en is set to 1 and this field is: 0 then updates occur once per field 1 then updates do not occur
4 rc_afterclamp_update_en	After clamp update enable Enables the after clamp line-by-line updates
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 rl_resetoffset	Reset offset Reset the integrator in the DC offset block
1 rl_disoffset	Disable offset Disables the DC offset output.
0 rh_shortframe	Short frame In normal operation, this field should be cleared to 0.

60.6.12 DC Offset (VDEC_DCOFF)

Address: 400C_8000h base + 78h offset = 400C_8078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									rl_dcoffsetP			rl_linemeasure_dis	Reserved	rl_dcoffsetI	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_DCOFF field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 rl_dcoffsetP	DC offset proportional gain Proportional gain of DC offset calculation
3 rl_linemeasure_dis	Line measure disable Disable line-by-line measurement for input clamp/DC-offset block
2 Reserved	This field is reserved.
1–0 rl_dcoffsetI	DC offset integrator gain Integrator gain of DC-offset calculation

60.6.13 Chroma Swap, Invert, and Debug (VDEC_CSID)

Address: 400C_8000h base + 84h offset = 400C_8084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								rd_bypasshilbert	Reserved			rd_nopalhue	rd_invcb	rd_invcr	rd_swapcrcb
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_CSID field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 rd_bypasshilbert	Bypass hilbert Disable the hilbert filter in the chroma demodulator (debug only)
6–4 Reserved	This field is reserved.
3 rd_nopalhue	No PAL hue 1 = disable the hue function when in PAL mode.
2 rd_invcb	Invert Cb Invert Cb output
1 rd_invcr	Invert Cr Invert Cr output
0 rd_swapcrcb	Swap Cr CB Swap Cr and Cb outputs.

60.6.14 Cb Gain (VDEC_CBGN)

Address: 400C_8000h base + 88h offset = 400C_8088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rd_cbgain							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_CBGN field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rd_cbgain	Cb gain Gain of Cb output. Nominal value is 0x80.

60.6.15 Cr Gain (VDEC_CRGN)

Address: 400C_8000h base + 8Ch offset = 400C_808Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rd_crgain							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_CRGN field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rd_crgain	Cr gain Gain of Cr output. Nominal value is 0x80.

60.6.16 Contrast (VDEC_CNTR)

Address: 400C_8000h base + 90h offset = 400C_8090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rd_lumagain							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_CNTR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rd_lumagain	Contrast This is equivalent to contrast. The pivot point is 0. Other contrast gains use a pivot point of mid luma. Nominal gain of 1 is 0x80.

60.6.17 Brightness (VDEC_BRT)

Address: 400C_8000h base + 94h offset = 400C_8094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rc_blacklevel							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_BRT field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rc_blacklevel	Brightness This is equivalent to brightness. Accepts a signed value from -128 to +127 (0x80 to 0x7f).

60.6.18 Hue (VDEC_HUE)

Address: 400C_8000h base + 98h offset = 400C_8098h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rd_ch_thresh							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_HUE field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rd_ch_thresh	Hue

60.6.19 Chroma Burst Threshold (VDEC_CHBTH)

Address: 400C_8000h base + 9Ch offset = 400C_809Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rd_ch_thresh							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_CHBTH field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rd_ch_thresh	Chroma burst threshold This is the level above which the chroma burst must be in order to process chroma. Below this value no chroma is assumed. Accepts unsigned number from 0 to 127.

60.6.20 Sharpness Improvement (VDEC_SHPIMP)

Address: 400C_8000h base + A4h offset = 400C_80A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rd_slope				rd_peak			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_SHPIMP field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 rd_slope	Slope Slope value is a signed number. 0x0 No effect 0x1 - 0x7 Positive slope compensation 0xf - 0x8 Negative slope compensation
3–0 rd_peak	Peak Added luma sharpness 0 No sharpness increase 15 Maximum sharpness increase

60.6.21 Chroma PLL and Input Mode (VDEC_CHPLLIM)

Address: 400C_8000h base + A8h offset = 400C_80A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									rd_chlock_atten			rd_locked_force	rd_inputcables		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_CHPLLIM field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 rd_chlock_atten	Chroma lock attenuation Sets the attenuation for the chroma lock detector.
3 rd_locked_force	Locked force Force chroma to always think its locked. Used in debug mode.
2–0 rd_inputcables	Input cables Set the input mode. Added luma sharpness. 000 CVBS 001 S-Video 010 Component 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved

60.6.22 Video Mode (VDEC_VIDMOD)

Address: 400C_8000h base + ACh offset = 400C_80ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								Pal	F443	Secam	m625	ch_locked	chroma	Hlocked	havesignal
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

VDEC_VIDMOD field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 Pal	PAL detected 0 PAL modulation not detected. 1 PAL modulation detected.
6 F443	4.43MHz chroma detected

Table continues on the next page...

VDEC_VIDMOD field descriptions (continued)

Field	Description
	0 No chroma carrier in the 4.43MHz range is detected. 1 A chroma carrier in the 4.43MHz range is detected.
5 Secam	SECAM detected 0 SECAM modulation not detected. 1 SECAM modulation detected.
4 m625	625 mode 0 The signal is not in 625 line mode. 1 The signal is in 625 line mode.
3 ch_locked	Chroma locked 0 Not locked to the chroma carrier 1 Locked to the chroma carrier
2 chroma	Chroma carrier detected 0 A chroma carrier is not present. 1 A chroma carrier is present.
1 Hlocked	Hsync locked 0 Not locked to the Hsync 1 Locked to the Hsync
0 havesignal	Have signal 0 A valid video signal is not detected. 1 A valid video signal is detected.

60.6.23 Video Status (VDEC_VIDSTS)

Address: 400C_8000h base + B0h offset = 400C_80B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										Reserved		0	Vcrdetect	Nonarith3D	Nonarith
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_VIDSTS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 Reserved	This field is reserved.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Vcrdetect	VCR detected High when a VCR signal is detected
1 Nonarith3D	Nonarithmetic 3D ratio High when an invalid nonarithmetic ratio for 3D comb is detected

Table continues on the next page...

VDEC_VIDSTS field descriptions (continued)

Field	Description
0 Nonarith	Invalid nonarithmetic ratio High when an invalid nonarithmetic ratio detected

60.6.24 Noise Detector (VDEC_NOISE)

Address: 400C_8000h base + B4h offset = 400C_80B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								Noise							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_NOISE field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 Noise	Noise detector Noise detector output. Value is proportional to noise. 0 = no noise. 1-255 = noise value detected.

60.6.25 Standards and Debug (VDEC_STDDBG)

Address: 400C_8000h base + B8h offset = 400C_80B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								rd_fc_maual	ntscj	force_2dntsc443	Reserved	force_havesignal	Reserved	standard_ filter	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_STDDBG field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 rd_fc_maual	Manual carrier frequency Enable manual Fc carrier frequency instead of hardcoded internal frequencies.
6 ntscj	NTSC keep pedestal When this field is set to 1, detection of NTSC will not remove the pedestal level.
5 force_2dntsc443	Force 2D NTSC 443 For the use of the 2D comb filter in NTSC 443 mode
4 Reserved	This field is reserved.
3 force_havesignal	Force have signal Override the video signal detector so as to appear to always have a valid video signal.
2 Reserved	This field is reserved.
1–0 standard_filter	Standard filter This field should be set to 3. Lower values cause faster standards detect.

60.6.26 Manual Override (VDEC_MANOVR)

NOTE

Pink cast over images in PAL mode can be got rid of by writing a value of 0xFD in this register. Upper nibble 0xF overrides auto detect of PAL and SECAM, while lower nibble 0xD selects PAL manually.

Address: 400C_8000h base + BCh offset = 400C_80BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								pal_override	f443_override	secam_override	line625_override	pal_manual	four43_manual	secam_manual	manual_625
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_MANOVR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 pal_override	PAL override 0 No override 1 Override PAL mode auto detection
6 f443_override	443 override 0 No override 1 Override 443 mode auto detection
5 secam_override	SECAM override 0 No override 1 Override SECAM mode auto detection
4 line625_override	Line 625 override

Table continues on the next page...

VDEC_MANOVR field descriptions (continued)

Field	Description
	0 No override 1 Override 625-line mode auto detection
3 pal_manual	PAL manual override 0 No override 1 Override manual setting of PAL mode with auto mode
2 four43_manual	443 manual override 0 No override 1 Override manual setting of 443 mode with auto mode
1 secam_manual	SECAM manual 0 No override 1 Manual setting of SECAM mode with auto mode is overridden
0 manual_625	Manual 625 0 No override 1 Manual setting of 625 line mode with auto mode is overridden

60.6.27 VSync and Signal Thresholds (VDEC_VSSGTH)

Address: 400C_8000h base + C8h offset = 400C_80C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								rh_vsynclength				0	nosigthresh		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_VSSGTH field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–4 rh_vsynclength	Vsync length Sets the length of the vsync detector, which is the number of consecutive vsyncs in the same place required to set a new phase/freq.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–0 nosigthresh	No signal threshold Sets the no-signal detection threshold

60.6.28 Debug Framebuffer (VDEC_DBGFBH)

Address: 400C_8000h base + D0h offset = 400C_80D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														clamp_delayH	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_DBGFBH field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1–0 clamp_delayH	Clamp delay high Sets the delay in 13.5MHz clocks for the application of the up/down pulses for the AFE DC clamp control. This is used to move the pulses to an area offscreen where it does not cause interference.

60.6.29 Debug Framebuffer 2 (VDEC_DBGFBL)

Address: 400C_8000h base + D4h offset = 400C_80D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								clamp_delayL							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_DBGFBL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 clamp_delayL	Clamp delay low Sets the delay in 13.5MHz clocks for the application of the up/down pulses for the AFE DC clamp control. This is used to move the pulses to an area offscreen where it does not cause interference.

60.6.30 H Active Start (VDEC_HACTS)

Address: 400C_8000h base + D8h offset = 400C_80D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								ro_hactivestart							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_HACTS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 ro_hactivestart	H active start Programs the start of the Hactive output. Measured in pixels after the Hsync falling edge.

60.6.31 H Active End (VDEC_HACTE)

Address: 400C_8000h base + DCh offset = 400C_80DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								ro_hactiveend							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_HACTE field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 ro_hactiveend	H active end Programs the end of the Hactive output. Measured in pixels before the Hsync falling edge. The longer total line of 625 line modes is automatically taken care of.

60.6.32 V Active Start (VDEC_VACTS)

Address: 400C_8000h base + E0h offset = 400C_80E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								ro_vactivestart							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_VACTS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 ro_vactivestart	V active start Programs the start of the Vactive output. Measured in half lines after the Vsync

60.6.33 V Active End (VDEC_VACTE)

Address: 400C_8000h base + E4h offset = 400C_80E4h

Bit	31–8	7–0
R	0	ro_vactiveend
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

VDEC_VACTE field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 ro_vactiveend	V active end Programs the end of the Vactive output. Measured in half lines before the Vsync.

60.6.34 HSync Tip (VDEC_HSTIP)

Address: 400C_8000h base + ECh offset = 400C_80ECh

Bit	31–8	7–0
R	0	rh_tipgate_start
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

VDEC_HSTIP field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rh_tipgate_start	Tip gate start Set the position in pixels of the start of the Hsync tip gate, which is always 32 pixels. This is used in the after clamp.

60.6.35 Bluescreen Cr (VDEC_BLSCRCR)

Address: 400C_8000h base + F8h offset = 400C_80F8h

Bit	31–8	7–0
R	0	bluescreen_y
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

VDEC_BLSCRCR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 bluescreen_y	Bluescreen Y The Y output value when in bluescreen mode, which is set when there is no valid video signal input detected.

60.6.36 Bluescreen Cb (VDEC_BLSCRCB)

Address: 400C_8000h base + FCh offset = 400C_80FCh

Bit	31–8	7–0
R	0	bluescreen_cb
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

VDEC_BLSCRCB field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 bluescreen_cb	Blue screen Cb The Cb output value when in blue screen mode, which is set when there is no valid video signal input detected.

60.6.37 Luma AGC Control 2 (VDEC_LMAGC2)

Address: 400C_8000h base + 104h offset = 400C_8104h

Bit	31–8	7–0
R	0	ragc_target
W		
Reset	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

VDEC_LMAGC2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 ragc_target	AGC target Sets the sync tip to black level target value for the AGC.

60.6.38 Chroma AGC Control 2 (VDEC_CHAGC2)

Address: 400C_8000h base + 10Ch offset = 400C_810Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								rd_chagc_target							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_CHAGC2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 rd_chagc_target	Chroma AGC target Sets the burst target height for the chroma AGC.

60.6.39 Minimum Threshold (VDEC_MINTH)

Address: 400C_8000h base + 114h offset = 400C_8114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								minthresh							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_MINTH field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 minthresh	Minimum threshold Sets the threshold for the minimum filter in the Hsync PLL. Larger values improve noise immunity.

60.6.40 Vertical Lines High (VDEC_VFRQOH)

Address: 400C_8000h base + 11Ch offset = 400C_811Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												vfreqo			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

VDEC_VFRQOH field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–0 vfreqo	Vertical frequency Number of half vertical lines detected in video field. If the number is odd then signal is interlaced video.

60.6.41 Vertical Lines Low (VDEC_VFRQOL)

Address: 400C_8000h base + 120h offset = 400C_8120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								vfreqo							
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

VDEC_VFRQOL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 vfreqo	Vertical frequency Number of half vertical lines detected in video field. If the number is odd then signal is interlaced video.

Chapter 61
System JTAG Controller (SJC)

61.1 Introduction

System JTAG Controller (SJC) provides the security authentication for debug access to the chip. It is accessible through the JTAG Access Port (AP) of the ARM DAP.

This figure shows SJC connections to external contacts and other blocks.

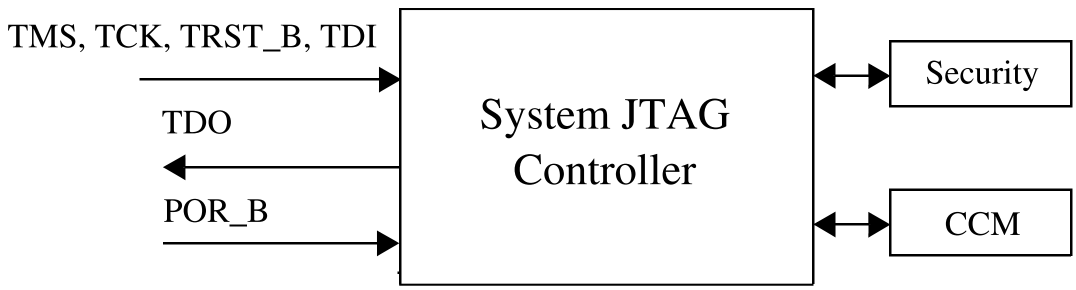


Figure 61-1. SJC connections

61.2 Programmable registers

This section lists additional registers to the standard accessible JTAG registers (per IEEE1149.1 standard). The following registers are accessed using the ExtraDebug mechanism, controlled by the ENABLE_ExtraDebug IR instruction.

NOTE

SJC registers are only accessible by JTAG interface and are not memory mapped to processor address space.

SJC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5	Security Status Register (SJC_SSR)	32	R	0000_0600h	61.2.1/3434

61.2.1 Security Status Register (SJC_SSR)

Address: 0h base + 5h offset = 5h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			RSSTAT		SJM		FT	Reserved	Reserved	EBG	EBF	SWE	SWF	Reserved	Reserved
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

SJC_SSR field descriptions

Field	Description
31–13 Reserved	This field is reserved.
12–11 RSSTAT	Response status 00 Response not entered 01 Response entered but not verified 10 Response entered and is incorrect 11 Response is correct

Table continues on the next page...

SJC_SSR field descriptions (continued)

Field	Description
10–9 SJM	SJC mode. Secure JTAG mode, as set by external fuses. These bits do not include the setting of the BSF fuse. 00 No debug (Mode 1) 01 Secure JTAG (Mode 2) 10 Reserved 11 JTAG enabled (Mode 3)
8 FT	Fuse type 0 E-fuse technology 1 Laser fuse technology
7 Reserved	This field is reserved.
6 Reserved	This field is reserved.
5 EBG	External boot granted. 1 Granted 0 Not granted
4 EBF	External Boot fuse. Status of the external boot disable fuse 0 Intact - external boot is allowed 1 Burned - external boot is disabled
3 SWE	SW JTAG enable status 1 Enabled 0 Disabled
2 SWF	Software JTAG enable fuse. Status of the no software disable JTAG fuse 0 Intact - software enable possible 1 Intact - no software enable possible
1 Reserved	This field is reserved.
0 Reserved	This field is reserved.

Chapter 62
System Bus Interconnect

62.1 Overview

The CoreLink Network Interconnect (NIC301) is a second generation highly configurable IP component that enables the creation of a complete high performance, optimized AMBA-compliant network infrastructure. It is based around a high-performance AXI crossbar switch known as the AXI bus matrix.

Recall the AMBA-AXI protocol is ARM's Advanced e*X*ensible *I*nterface, the third generation AMBA interface definition. It is targeted for high performance, high frequency system designs and is suitable for high bandwidth, low latency data transfers. The key features of the AXI protocol include:

- Burst-based, split transaction bus
- Separate address/control and data phases
- Ability to issue multiple outstanding addresses
- Out-of-order transaction completion
- Five independent channels
    - Read address
    - Write address
    - Read data
    - Write data
    - Write response
- Each channel uses 2-way handshake mechanism (VALID, READY control signals)

The possible configurations for the CoreLink Network Interconnect can range from a single bridge component, for example an AHB to AXI protocol bridge, to a complex infrastructure that consists of up to 128 masters and 64 slaves of a combination of different AMBA protocols.

As the device includes a large number of bus master and slave devices implementing both AXI and AHB bus protocols with 32- or 64-bit datapath widths and operating at multiple synchronous frequencies, the NIC301 is an ideal system bus "fabric" and provides the needed hardware interconnect matrix for the device.

A CoreLink Network Interconnect configuration can consist of multiple switches with many topology options. Figure 62-1 shows a top-level block diagram of the CoreLink Network Interconnect that contains:

- Multiple switches
- Multiple AMBA Slave Interface Blocks (ASIBs)
- Multiple AMBA Master Interface Blocks (AMIBs)

Note in the context of the NIC301, a *system bus master* connects to an AMBA *Slave* Interface (ASIB) and a *bus slave* connects to an AMBA *Master* Interface Block (AMIB).

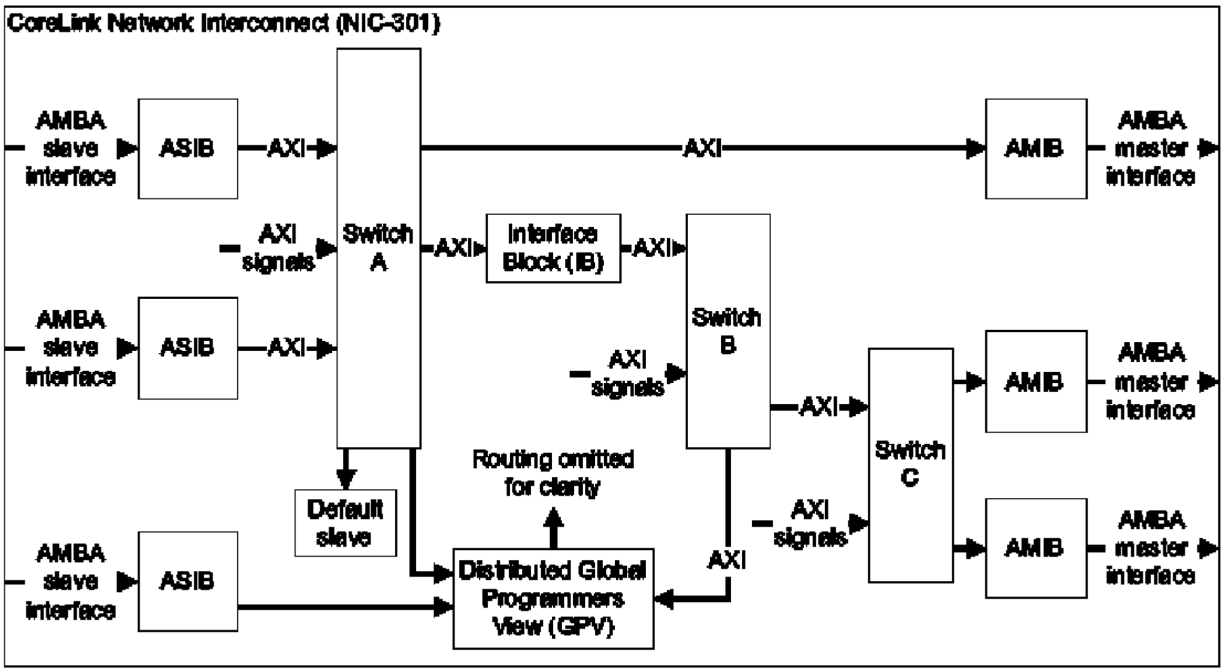


Figure 62-1. NIC301 Block Diagram

The table below provides an overview of the characteristics of the bus masters and slaves.

Table 62-1. Bus Master and Slave Interface Characteristics

Module	Type	interface Protocol	Interface Ports	Datapath Width	Operating Speed
Cortex-A5	Master	AXI	1	64b	High-speed Core
Cortex-M4	Master	AHB-Lite	2	64b	Low-speed Core = Platform
DMA2x	Master	AHB-Lite	2[1]	64b	Platform
DCU	Master	AXI	2[1]	64b	Platform
USB OTG	Master	AHB-Lite	2[1]	32b	Platform/2
Open VG	Master	AXI	1	64b	Platform
ENET	Master	AHB-Lite	2[2]	32b	Platform/2
VIU3	Master	AHB-Lite	1	64b	Platform
eSDHC	Master	AHB-Lite	2[1]	32b	Platform
NFC	Master	AHB-Lite	1	32b	Platform
CAAM	Master	AXI	1	32b	Platform
Debug Access Port (DAP)	Master	AHB-Lite	1	32b	Platform
ROM	Slave	AHB-Lite	1	64b	Platform
FlexBus	Slave	AHB-Lite	1	32b	Platform
QuadSPI, RLE	Slave	AHB-Lite	2[1]	64b	Platform
OCRAM	Slave	AXI	3[1]	64b	Platform
CM4_TCM	Slave	AHB-Lite	1	64b	Platform
SDRAMC (DDR)	Slave	AXI	2	64b	Platform
SecureRAM	Slave	AXI	1	32b	Platform
PBRIDGE (Peripheral Bridges)	Slave	AHB-Lite	2	64b	Platform

1. For these master and slaves, the multiple interface ports are due to multiple instances of the given module, for example, there are two DCU modules and two DMA2x modules. etc.
2. One of the ENET master ports (ENET1) is statically muxed with the master port from the MLB50 module which supports the MediaLB protocol for transfers on the MOST®data network. From the perspective of the NIC301 switch, the combined ENET1/MLB50 port appears as a single bus master.

62.2 Features

The CoreLink Network Interconnect NIC301 is a highly configurable infrastructure component that includes the following features:

- 1-128 AXI or AHB-Lite slave interfaces for bus master connections
- 1-64 master interfaces that can be AXI, AHB-Lite, APB2, or APB3 for bus slave connections
- Single-cycle arbitration Full pipelining to prevent master stalls
- Programmable control for FIFO transaction release

- Multiple switch networks
- AXI or AHB-Lite masters and slaves
    - Address width of 32-64 bits
    - Data width of 32, 64, 128, or 256 bits
- Non-contiguous APB slave address map for a single master interface
- Independent widths of user-defined sideband signals for each channel
- Global Programmers View (GPV) for the entire infrastructure, configurable for customizing the memory mapped visibility
- Highly flexible timing closure options

A very important attribute of the entire system bus fabric supported by the NIC301 and its split transaction protocol is the ***optimization for multi-master data bandwidth*** (versus data transfer latencies). *The device architecture implements a high performance microprocessor architecture where system performance is maximized by making effective use of the processor local memories (L1 and L2 caches, tightly coupled memories) and 32-byte cache line size 4-beat burst data transfers on the system bus fabric*. In environments and application spaces where these runtime characteristics are *not met*, data latency concerns can degrade system performance substantially.

62.3 NIC301 Physical Structure and Programming Model

62.3.1 NIC301 Physical Structure

The port assignments and NodeNumbers are of interest in the this section's discussion on the physical structure of the NIC301. The NodeNumbers are associated with the NIC301's Global Programmers View of the control and configuration registers. See Table 62-2

Table 62-2. NIC301 Port Assignments and NodeNumbers

Module	Type	PortNumber	NodeNumber
Cortex-M4 Code	Master	m0	66
Cortex-M4 System	Master	m1	67
Cortex-A5	Master	m2	68
DMA2x0	Master	m3	69
DMA2x1	Master	m4	70
DCU0	Master	m5	71
USB OTG0	Master	m6	72
Open VG	Master	m7	73

Table continues on the next page...

Table 62-2. NIC301 Port Assignments and NodeNumbers (continued)

Module	Type	PortNumber	NodeNumber
ENET0	Master	m8	74
ENET1 / MLB50	Master	m9(a,b)	75
VIU3	Master	m10	76
USB OTG1	Master	m11	77
DCU1	Master	m12	78
eSDHC0	Master	m13	79
eSDHC1	Master	m14	80
NFC	Master	m15	81
CAAM	Master	m16	82
Debug Access Port (DAP)	Master	m17	83
ROM	Slave	s0	2
FlexBus	Slave	s0	2
PBRIDGE0	Slave	s0	2
CM4_TCMx	Slave	s1	3
PBRIDGE1	Slave	s1	3
QuadSPI0	Slave	s2	4
OCRAM0_sys	Slave	s3	5
OCRAM1_sys	Slave	s4	6
OCRAM2_gfx	Slave	s5	7
SecureRAM	Slave	s6	8
QuadSPI1	Slave	s7	9
RLE	Slave	s7	9
SDRAMC (DDR0)	Slave	s8	10
SDRAMC (DDR1)	Slave	s9	11

NOTE

Masters that can access S8 (DDR0):

- M0, M1 (CM4)
- M5 (DCU0)
- M6 (USB)
- M10 (VIU2)
- M3, M8, M13, M15, M17 (DMA0, ENET0, eSDHC0, NFC, DBG)

NOTE

Masters that can access S9 (DDR1) are:

- M2 (CA5)
- M7 (Open VG)

- M12 (DCU1)
- M4, M9a, M9b, M11, M14, M16 (DMA1, ENET1, MLB, USB, eSDHC2, CAAM)

NOTE

There are a few restrictions on the masters that can access S6 port. The following masters cannot access S6 (SecureRAM):

- M5 (DCU0)
- M7 (Open VG)
- M10 (VIU2)
- M12 (DCU1)

Note the multiple shared NIC301 port connections (s0, s1, s7) to slave modules. For these shared slave connections, the device dual core platform includes “port splitters” to route the data transfers to the appropriate target module.

The internal structure of the device NIC301 implementation is shown in the figure below. For this complex system bus fabric, the structure balances the masters and slaves based on the expected data traffic and by grouping modules with similar bus protocols. The structure includes 3 master switch concentrators feeding, along with 6 masters with direct connections, into the main bus switch (switch3). Additionally, there is a 1-to-4 bus splitter on the slave side of the main switch. The resulting structure represents a trade-off balancing bus switch size, MHz timing and system bandwidth considerations.

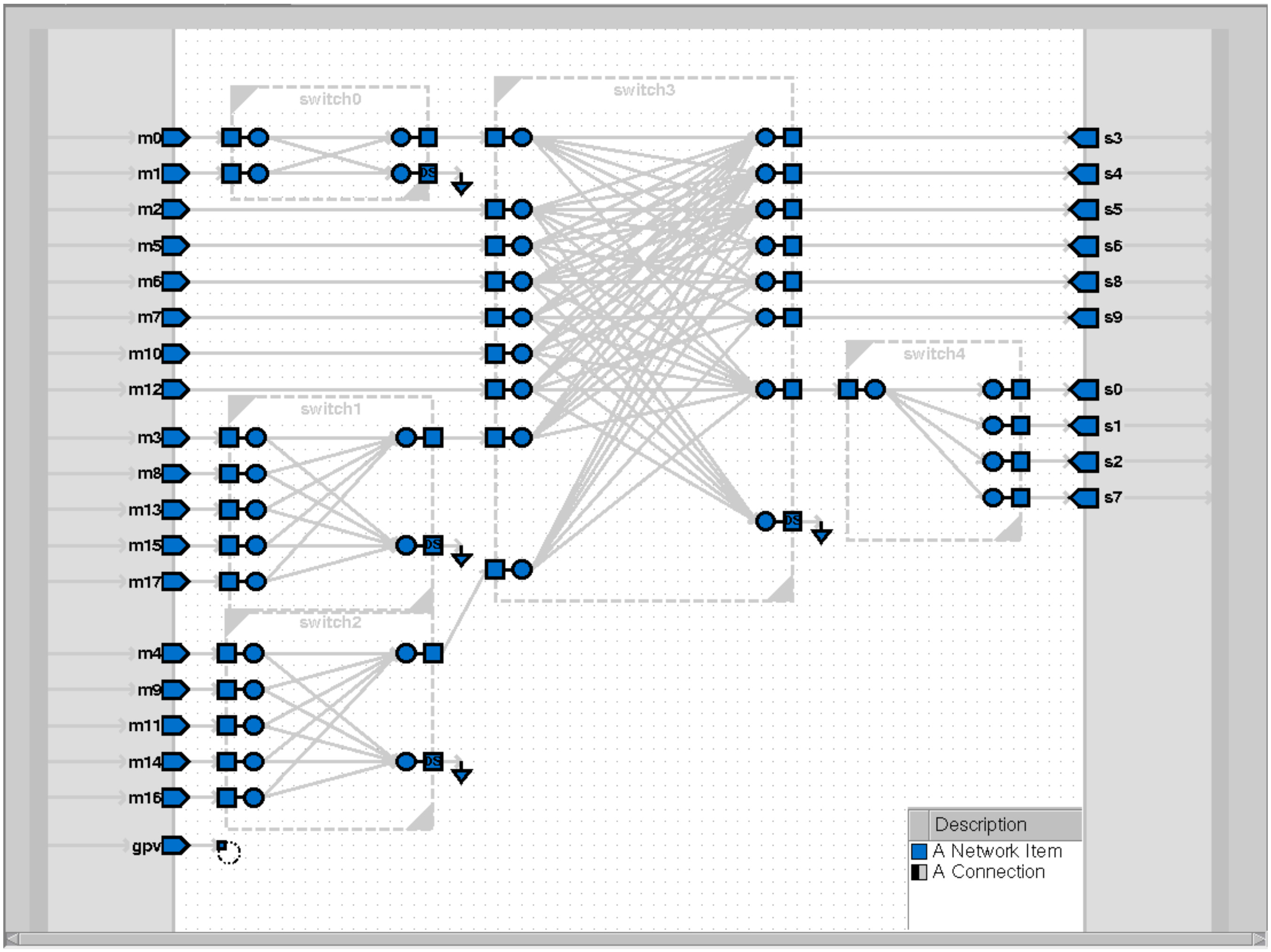


Figure 62-2. NIC301 Physical Block Diagram

62.3.2 NIC301 Programming Model

The individual AMBA master and slave interface blocks contained in the NIC301 each support a number of configuration registers for defining the operation of the bus switch. This section provides details on the memory map associated with the NIC301's GPV address space; details on the operation of specific control fields within these GPV registers can be found in the appropriate ARM documentation. The device implementation only allows privilege mode references; attempted references in user mode are error terminated.

As previously noted, the NIC301 is a very highly configurable IP component. To support the wide range of possible NIC301 configurations, the GPV is defined by the maximum configuration, and occupies a 1 Mbyte address space consisting of 256 4 Kbyte address spaces. Specifically, this 1 Mbyte address space consists of the following concatenation of register blocks:

- Address control register block
- Peripheral ID register block
- 64 master interface (AMIB) register blocks
- 128 slave interface (ASIB) register blocks
- 62 internal interface (IB) register blocks

Unfortunately, the replicated register blocks are *very sparsely populated*, both in terms of address locations used as well as the total number of defined register bits. As examples, the AMIB register block only includes 14 register bits per 4 Kbyte space; the ASIB includes 21 registers bit per 4 Kbyte space and the IB 12 register bits per 4 Kbyte space. The internal interface (IB) register blocks are *not used in the device*, but are partially supported in the architectural definition of the GPV address compression scheme.

The device uses an address compression technique to reduce the NIC301 GPV address space from 1 MB down to 32 Kbytes, which is then mapped to 8 consecutive 4 Kbyte "slots" managed by the PBRIDGE0 bus controller. This space is located at addresses 0x4000_8000 - 0x4000_FFFF in the device memory map.

To support the required 32:1 address space compression, the GPV's 20-bit 1 Mbyte address must be reduced to a 15-bit 32 Kbyte address for the PBRIDGE. For the GPV, the register base offset address is calculated as 4096 * NodeNumber; in general, the PBRIDGE register offset address is defined as 256 * NodeNumber with special handling of ADDR[7] providing the final bit of address compression. See Table 62-3 for the association between the GPV and PBRIDGE register address offsets. Recall the NodeNumbers were defined in Table 62-2

Table 62-3. NIC301 GPV → PBRIDGE Base Address Offset Compression

GPV Register Block	GPV Base Offset	PBRIDGE Base Offset	NodeNumber	GPV Address Offset	PBRIDGE Address Offset
Address Control	0x0_0000	0x0000	0	4096 * NodeNumber	256 * NodeNumber
Peripheral ID	0x0_1000	0x0100	1	4096 * NodeNumber	256 * NodeNumber
Master (AMIB) Interface [0-31]	0x0_2000 - 0x2_1000	0x0200 - 0x2100	2-33	4096 * NodeNumber	256 * NodeNumber
Master (AMIB) Interface [32-63]	0x2_2000 - 0.x4_1000	Not supported	34-65	4096 * NodeNumber	Not supported
Slave (ASIB) Interface [0-63]	0x4_2000 - 0x8_1000	0x2200 - 0x6100	66-129	4096 * NodeNumber	256 * (NodeNumber - 32)
Slave (ASIB) Interface [64-127]	0x8_2000 - 0xC_1000	Not supported	130-193	4096 * NodeNumber	Not Supported
Internal (IB) Interface [0-29]	0xC_2000 - 0xD_F000	0x6200 - 0x7F00	194 - 223	4096 * NodeNumber	256 * (NodeNumber - 96)
Internal (IB) Interface [30-61]	0xE_0000 - 0xF_F000	Not supported	224-255	4096 * NodeNumber	Not Supported

The final details associated with the GPV address space compression involves the behavior of ADDR[7]. For register offsets < 0x80, the GPV and PBRIDGE addresses are identical. Fortunately, the number of register offsets not meeting this condition are small in number. See Table 62-4 for the details on the required compression between GPV_ADDR[11:0] and PBRIDGE_ADDR[7:0].

Table 62-4. NIC301 GPV and PBRIDGE Register Offset Addresses

GPV Register Block	GPV_ADDR [11:0]	PBRIDGE_ADDR [7:0]	Register
All	< 0x080	< 0x80	Any register with these offsets
Address Control	0x080	0x80	security30
	0x084	0x84	security31 (Max GPV address = 0x084)
Peripheral ID	0xFD0	0xD0	Peripheral ID4
	0xFD4	0xD4	Peripheral ID5
	0xFD8	0xD8	Peripheral ID6
	0xFDC	0xDC	Peripheral ID7
	0xFE0	0xE0	Peripheral ID0
	0xFE4	0xE4	Peripheral ID1
	0xFE8	0xE8	Peripheral ID2
	0xFEC	0xEC	Peripheral ID3
	0xFF0	0xF0	Component ID0
	0xFF4	0xF4	Component ID1
	0xFF8	0xF8	Component ID2
	0xFFC	0xFC	Component ID3
Master (AMIB) Interface [0-31]	0x108	0x88	fn_mod
Slave (ASIB) Interface [0-63]	0x100	0x80	read_qos
	0x104	0x84	write_qos
	0x108	0x88	fn_mod
Internal (IB) Interface [0-29]	0x108	0x88	fn_mod

Combining all the information detailed in Table 62-3 and Table 62-4, the complete definition of the device memory map for the NIC301 GPV registers is presented in Table 62-5.

Table 62-5. NIC301 GPV Registers and PBRIDGE Access Addresses

GPV Register Block	GPV_ADDR [11:0]	Device ADDR	Register
AddressControl	0x0_008 + 4*n	0x4000_8008 + 4*n	Slave Security <n>

Table continues on the next page...

Table 62-5. NIC301 GPV Registers and PBRIDGE Access Addresses (continued)

GPV Register Block	GPV_ADDR [11:0]	Device ADDR	Register
PeripheralID	0x0_1FD0	0x4000_81D0	Peripheral ID4
	0x0_1FD4	0x4000_81D4	Peripheral ID5
	0x0_1FD8	0x4000_81D8	Peripheral ID6
	0x0_1FDC	0x4000_81DC	Peripheral ID7
	0x0_1FE0	0x4000_81E0	Peripheral ID0
	0x0_1FE4	0x4000_81E4	Peripheral ID1
	0x0_1FE8	0x4000_81E8	Peripheral ID2
	0x0_1FEC	0x4000_81EC	Peripheral ID3
	0x0_1FF0	0x4000_81F0	Component ID0
	0x0_1FF4	0x4000_81F4	Component ID1
	0x0_1FF8	0x4000_81F8	Component ID2
	0x0_1FFC	0x4000_81FC	Component ID3
Master (AMIB) Interface, NodeNumbers = [2-33]	4096 * NodeNumber + 0x008	0x4000_8008 + 256 * NodeNumber	fn_mod_bm_iss
	4096 * NodeNumber + 0x020	0x4000_8020 + 256 * NodeNumber	sync_mode
	4096 * NodeNumber + 0x024	0x4000_8024 + 256 * NodeNumber	fn_mod2
	4096 * NodeNumber + 0x040	0x4000_8040 + 256 * NodeNumber	wr_tidemark
	4096 * NodeNumber + 0x044	0x4000_8044 + 256 * NodeNumber	ahb_cntl
	4096 * NodeNumber + 0x108	0x4000_8088 + 256 * (NodeNumber - 32)	fn_mod
Slave (ASIB) Interface, NodeNumbers = [66-129]	4096 * NodeNumber + 0x020	0x4000_8020 + 256 * (NodeNumber - 32)	sync_mode
	4096 * NodeNumber + 0x024	0x4000_8024 + 256 * (NodeNumber - 32)	fn_mod2
	4096 * NodeNumber + 0x028	0x4000_8028 + 256 * (NodeNumber - 32)	fn_modl_ahb
	4096 * NodeNumber + 0x040	0x4000_8040 + 256 * (NodeNumber - 32)	wr_tidemark
	4096 * NodeNumber + 0x100	0x4000_8080 + 256 * (NodeNumber - 32)	read_qos
	4096 * NodeNumber + 0x104	0x4000_8084 + 256 * (NodeNumber - 32)	write_qos
	4096 * NodeNumber + 0x108	0x4000_8088 + 256 * (NodeNumber - 32)	fn_mod

Table continues on the next page...

Table 62-5. NIC301 GPV Registers and PBRIDGE Access Addresses (continued)

GPV Register Block	GPV_ADDR [11:0]	Device ADDR	Register
Internal (IB) Interface, NodeNumbers = [194-223]	4096 * NodeNumber + 0x008	0x4000_8008 + 256 * (NodeNumber - 96)	fn_mod_bm_iss
	4096 * NodeNumber + 0x020	0x4000_8020 + 256 * (NodeNumber - 96)	sync_mode
	4096 * NodeNumber + 0x024	0x4000_8024 + 256 * (NodeNumber - 96)	fn_mod2
	4096 * NodeNumber + 0x040	0x4000_8040 + 256 * (NodeNumber - 96)	wr_tidemark
	4096 * NodeNumber + 0x108	0x4000_8088 + 256 * (NodeNumber - 96)	fn_mod

62.3.3 NIC301 Bus Arbitration

The NIC301 switch implements a two level arbitration mechanism that is partially programmable.

There is a 4-bit *Quality of Service (QoS)* priority that can be assigned (both a default value at reset plus a software programmed value) to each master port. Within the NIC301, the arbitration logic uses the QoS values for a first level fixed priority scheme; the higher the QoS value, the higher the master's priority. For multiple requesting masters with the same QoS value, a second level least-recently-used (also known as a least-recently-granted) scheme is applied.

The NIC301 Technical Reference Manual states “At any arbitration node, a fixed priority exists for transactions with a different QoS. The highest value has the highest priority. If there are coincident transactions at an arbitration node with the same QoS that require arbitration, then the Network uses a Least Recently Used (LRU) algorithm.”

Given these arbitration capabilities, the device dual-core platforms defines a default set of QoS values that represents a “recommended system configuration”. The default QoS values for the NIC301's bus masters are shown in Table 62-6

Table 62-6. NIC301 Default QoS Arbitration Priorities

System Bus Master	Port Number	Default QoS
Cortex-M4 Core	m0	1
Cortex-M4 System	m1	2
Cortex-A5	m2	0
DMA2x0	m3	4
DMA2x1	m4	4

Table continues on the next page...

Table 62-6. NIC301 Default QoS Arbitration Priorities (continued)

System Bus Master	Port Number	Default QoS
DCU0	m5	14
USB OTG0	m6	11
Open VG	m7	12
ENET0	m8	10
ENET1 / MLB50	m9{a,b}	10
VIU3	m10	13
USB OTG1	m11	11
DCU1	m12	14
eSDHC0	m13	9
eSDHC1	m14	9
NFC	m15	3
CAAM	m16	5
Debug Access Port (DAP)	m17	15

Chapter 63
Miscellaneous Control Module (MCM)

63.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration information.

The Miscellaneous Control Module (MCM) provides a myriad of miscellaneous control functions.

63.1.1 Features

The MCM includes the following features:

- Program-visible information on the platform configuration and revision

63.2 Memory map/register descriptions

The memory map and register descriptions below describe the registers using byte addresses.

MCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_0008	Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)	16	R	00FFh	63.2.1/3450
E008_000A	Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)	16	R	00FFh	63.2.2/3451
E008_000C	Control Register (MCM_CR)	32	R/W	0000_0000h	63.2.3/3452

Table continues on the next page...

MCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_0010	Interrupt Status and Control Register (MCM_ISCR)	32	R	0002_0000h	63.2.4/3453
E008_0020	Fault address register (MCM_FADR)	32	R	Undefined	63.2.5/3456
E008_0024	Fault attributes register (MCM_FATR)	32	R	Undefined	63.2.6/3457
E008_0028	Fault data register (MCM_FDR)	32	R	Undefined	63.2.7/3459

63.2.1 Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)

PLASC is a 16-bit read-only register identifying the presence/absence of bus slave connections to the device's crossbar switch.

Address: E008_0000h base + 8h offset = E008_0008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ASC							
Write																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

MCM_PLASC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 ASC	Each bit in the ASC field indicates whether there is a corresponding connection to the crossbar switch's slave input port. 0 A bus slave connection to AXBS input port *n* is absent 1 A bus slave connection to AXBS input port *n* is present

63.2.2 Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)

PLAMC is a 16-bit read-only register identifying the presence/absence of bus master connections to the device's crossbar switch.

Address: E008_0000h base + Ah offset = E008_000Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								AMC							
Write																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

MCM_PLAMC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–0 AMC	Each bit in the AMC field indicates whether there is a corresponding connection to the AXBS master input port. 0 A bus master connection to AXBS input port *n* is absent 1 A bus master connection to AXBS input port *n* is present

63.2.3 Control Register (MCM_CR)

CR defines the arbitration and protection schemes for the two system RAM arrays.

Address: E008_0000h base + Ch offset = E008_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SRAMLWP	SRAMLAP		0	SRAMUWP	SRAMUAP		Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						Reserved	Reserved								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_CR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 SRAMLWP	SRAM_L Write Protect When this bit is set, writes to SRAM_L array generates a bus error.
29–28 SRAMLAP	SRAM_L arbitration priority Defines the arbitration scheme and priority for the processor and SRAM backdoor accesses to the SRAM_L array. 00 Round robin 01 Special round robin (favors SRAM backoor accesses over the processor) 10 Fixed priority. Processor has highest, backdoor has lowest 11 Fixed priority. Backdoor has highest, processor has lowest
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

MCM_CR field descriptions (continued)

Field	Description
26 SRAMUWP	SRAM_U write protect When this bit is set, writes to SRAM_U array generates a bus error.
25–24 SRAMUAP	SRAM_U arbitration priority Defines the arbitration scheme and priority for the processor and SRAM backdoor accesses to the SRAM_U array. 00 Round robin 01 Special round robin (favors SRAM backoor accesses over the processor) 10 Fixed priority. Processor has highest, backdoor has lowest 11 Fixed priority. Backdoor has highest, processor has lowest
23–10 Reserved	This field is reserved.
9 Reserved	This field is reserved.
8–0 Reserved	This field is reserved.

63.2.4 Interrupt Status and Control Register (MCM_ISCR)

The MCM_ISCR register defines the configuration and reports status for a number of core-related interrupt exception conditions. It includes the enable and status bits associated with the core's floating-point exceptions, bus errors associated with the core's cache write buffer, and events associated with the debug ETB module. The individual event indicators are first qualified with their exception enables and then logically summed to form an interrupt request sent to the core's NVIC.

Bits 15-8 are read-only indicator flags based on the processor's FPSCR register. Attempted writes to these bits are ignored. Once set, the flags remain asserted until software clears the corresponding FPSCR bit.

Address: E008_0000h base + 10h offset = E008_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FIDCE	0		FIXCE	FUFCE	FOFCE	FDZCE	FIOCE	0			CWBEE	Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIDC	0		FIXC	FUFC	FOFC	FDZC	FIOC	0			CWBER	DHREQ	0		
W												w1c				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_ISCR field descriptions

Field	Description
31 FIDCE	FPU input denormal interrupt enable 0 Disable interrupt 1 Enable interrupt
30–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 FIXCE	FPU inexact interrupt enable 0 Disable interrupt 1 Enable interrupt
27 FUFCE	FPU underflow interrupt enable 0 Disable interrupt 1 Enable interrupt
26 FOFCE	FPU overflow interrupt enable 0 Disable interrupt 1 Enable interrupt
25 FDZCE	FPU divide-by-zero interrupt enable 0 Disable interrupt 1 Enable interrupt
24 FIOCE	FPU invalid operation interrupt enable 0 Disable interrupt 1 Enable interrupt
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 CWBEE	Cache write buffer error enable Enables the generation of an interrupt in response to a bus error termination reported on a system bus transfer initiated from the cache's write buffer.

Table continues on the next page...

MCM_ISCR field descriptions (continued)

Field	Description
	0 Disable error interrupt 1 Enable error interrupt
19–16 Reserved	This field is reserved.
15 FIDC	FPU input denormal interrupt status This read-only bit is a copy of the core's FPSCR[IDC] bit and signals input denormalized number has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IDC] bit. 0 No interrupt 1 Interrupt occurred
14–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 FIXC	FPU inexact interrupt status This read-only bit is a copy of the core's FPSCR[IXC] bit and signals an inexact number has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IXC] bit. 0 No interrupt 1 Interrupt occurred
11 FUFC	FPU underflow interrupt status This read-only bit is a copy of the core's FPSCR[UFC] bit and signals an underflow has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[UFC] bit. 0 No interrupt 1 Interrupt occurred
10 FOFC	FPU overflow interrupt status This read-only bit is a copy of the core's FPSCR[OFC] bit and signals an overflow has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[OFC] bit. 0 No interrupt 1 Interrupt occurred
9 FDZC	FPU divide-by-zero interrupt status This read-only bit is a copy of the core's FPSCR[DZC] bit and signals a divide by zero has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[DZC] bit. 0 No interrupt 1 Interrupt occurred
8 FIOC	FPU invalid operation interrupt status This read-only bit is a copy of the core's FPSCR[IOC] bit and signals an illegal operation has been detected in the processor's FPU. Once set, this bit remains set until software clears the FPSCR[IOC] bit. 0 No interrupt 1 Interrupt occurred
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

MCM_ISCR field descriptions (continued)

Field	Description
4 CWBER	Cache write buffer error status Signals a data transfer from the core's cache write buffer was terminated with a bus error. This bit only sets when the corresponding enable bit (CWBEE) is set. The corresponding core fault address, attributes and write data are typically retrieved from the FADR, FATR, and FDR registers during the interrupt service routine before clearing the CWBER flag. 0 No error 1 Error occurred
3 DHREQ	Debug Halt Request Indicator Indicates that a debug halt request is initiated due to a ETB counter expiration, ETBCC[2:0] = 3b111 & ETBCV[10:0] = 11h0. This bit is cleared when the counter is disabled or when the ETB counter is reloaded. 0 No debug halt request 1 Debug halt request initiated
2–0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

63.2.5 Fault address register (MCM_FADR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting address is captured in the MCM_FADR register. The MCM logic supports capturing a single cache write buffer bus error event; if a subsequent error is detected before the captured error information has been read from the corresponding registers and the MCM_ISCR[CWBER] indicator cleared, the MCM_FATR[BEOVR] flag is set. However, no additional information is captured.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. Attempted writes to this location are terminated with an error.

Address: E008_0000h base + 20h offset = E008_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDRESS																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MCM_FADR field descriptions

Field	Description
31–0 ADDRESS	Fault address

63.2.6 Fault attributes register (MCM_FATR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting attributes are captured in the MCM_FATR register.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. Attempted writes to this location are terminated with an error.

Address: E008_0000h base + 24h offset = E008_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BEOVR	0														
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				BEMN				BEWT	0	BESZ		0		BEMD	BEDA
W																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MCM_FATR field descriptions

Field	Description
31 BEOVR	Bus error overrun Indicates if anothercache write buffer bus error is detected before system software has retrieved all the error information from the original event, this overrun flag is set. The window of time is defined from the detection of the original cache write buffer error termination until the MCM_ISCR[CWBER] is written with a 1 to clear it and rearm the capture logic. This bit is set by the hardware and cleared whenever software writes a 1 to the CWBER bit. 0 No bus error overrun 1 Bus error overrun occurred. The FADR and FDR registers and the other FATR bits are not updated to reflect this new bus error.
30–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 BEMN	Bus error master number Crossbar switch bus master number of the captured cache write buffer bus error. For this device, this value is always 0x1.
7 BEWT	Bus error write Indicates the type of system bus access when the error was detected. Since this logic is monitoring data transfers from the cache write buffer, this bit is always a logical one, signaling a write operation. 0 Read access 1 Write access
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 BESZ	Bus error size Indicates the size of the cache write buffer access when the error was detected. 00 8-bit access 01 16-bit access 10 32-bit access 11 Reserved
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 BEMD	Bus error privilege level Indicates the privilege level of the cache write buffer access when the error was detected. 0 User mode 1 Supervisor/privileged mode
0 BEDA	Bus error access type Indicates the type of cache write buffer access when the error was detected. This attribute is always a logical one signaling a data reference. 0 Instruction 1 Data

63.2.7 Fault data register (MCM_FDR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting data is captured in the MCM_FDR register.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. For byte and halfword writes, only the accessed byte lanes contain valid data; the contents of the other bytes are undefined. Attempted writes to this location are terminated with an error.

Address: E008_0000h base + 28h offset = E008_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MCM_FDR field descriptions

Field	Description
31–0 DATA	Fault data

Chapter 64 AHB-TrustZone Address Space Controller (AHB-TZASC)

64.1 Overview

The AHB-TZASC provides functionality equivalent to the (AXI) TZASC module. There are four instances of the basic AHB-TZASC functionality, each supporting 8 memory "region descriptors" with a programming model exactly equivalent to the (AXI) TZASC's capabilities.

64.2 Memory Map and Register Descriptions

This section provides an overview summary of the AHB-TZASC's programming model. See the (AXI) TZASC documentation for the specific register details. In particular, see the ARM technical reference manual, *DDI0431C_tzasc_tzc380_r0p1_trm.pdf*.

Each instance of the AHB-TZASC is allocated a 1 KB address space, so that all 4 instances can be supported in a single 4 KB slave peripheral slot (on-platform slot 16, based at address 0x4001_0000) of the PBRIDGE0 controller.

Table 64-1. AHB-TZASC Programming Model Memory Map

System Base Address Range	AHB-TZASC Programming Model Description
0x4001_0000 - 0x4001_03FF	FlexBus
0x4001_0400 - 0x4001_07FF	CM4's Tightly-Coupled Memory Backdoor Port
0x4001_0800 - 0x4001_0BFF	QuadSPI0
0x4001_0C00 - 0x4001_0FFF	QuadSPI1

The programming model for *each* instance is shown in Table 64-2.

Table 64-2. AHB-TZASC Programming Model Map Details

Instance Offset Address	ARM Register Name	Width (bits)	Access	Reset Value
0x000	configuration	32	RO	0x0000_1F07
0x004	action	32	RW	0x0000_0001
0x008	lockdown_range	32	RW	0x0000_0000
0x00C	lockdown_select	32	RW	0x0000_0000
0x010	int_status	2	RO	0x0000_0000
0x014	int_clear	32	WO	0x0000_0000
0x018 - 0x01C	– (reserved)	32	RAZ/WI	–
0x020	fail_address_low	32	RO	0x0000_0000
0x024	fail_address_high	32	RO	0x0000_0000
0x028	fail_control	32	RO	0x0000_0000

Instance Offset Address	ARM Register Name	Width (bits)	Access	Reset Value
0x02C	fail_id	32	RO	0x0000_0000
0x030	speculation_control	32	RW	0x0000_0000
0x034	security_inversion_en	32	RW	0x0000_0000
0x038 - 0x0FC	– (reserved)	32	RAZ/WI	–
Region Descriptor 0				
0x100	region_setup_low_0	32	RW	0x0000_0000
0x104	region_setup_high_0	32	RW	0x0000_0000
0x108	region_attributes_0	32	RW	0xC000_0000
0x10C	– (reserved)	32	RAZ/WI	–
Region Descriptor 1				
0x110	region_setup_low_1	32	RW	0x0000_0000
0x114	region_setup_high_1	32	RW	0x0000_0000
0x118	region_attributes_1	32	RW	0xC000_0000
0x11C	– (reserved)	32	RAZ/WI	–
...				
Region Descriptor 7				
0x170	region_setup_low_7	32	RW	0x0000_0000
0x174	region_setup_high_7	32	RW	0x0000_0000
0x178	region_attributes_7	32	RW	0xC000_0000
0x17C	– (reserved)	32	RAZ/WI	–
0x180 - 0x3FC	– (reserved)	32	RAZ/WI	–

Next, a simplified view of the combined register formats is presented in Figure 64-1. See the ARM TZASC documentation for register field details. Note, any specific register bits that provide different functionality compared to the (AXI) TZASC are highlighted and discussed in more detail below.

Figure 64-1. AHB-TZASC Register Overview Summary

Register configuration
Offset Instance_Base + 0x000 — Access: Privileged read-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1

Register action
Offset Instance_Base + 0x004 — Access: Privileged read/write

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register lockdown_range
Offset Instance_Base + 0x008 — Access: Privileged read/write

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register lockdown_select
Offset Instance_Base + 0x00C — Access: Privileged read/write

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Register int_status
Offset Instance_Base + 0x010 — Access: Privileged read-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*	*
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register int_clear
Offset Instance_Base + 0x014 — Access: Privileged write-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register fail_address_low
Offset Instance_Base + 0x020 — Access: Privileged read-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	add_status_low																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register fail_address_high
Offset Instance_Base + 0x024 — Access: Privileged read-only

R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register fail_control
Offset Instance_Base + 0x028 Access: Privileged read-only

R	0	0	0	0	0	0	0	*	0	0	*	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register fail_id
Offset Instance_Base + 0x02C Access: Privileged read-only

R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	id				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register speculation_control
Offset Instance_Base + 0x030 Access: Privileged read/write

R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register security_inversion_en
Offset Instance_Base + 0x034 Access: Privileged read/write

R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

For Region Descriptor n

Register region_setup_low_n
Offset Instance_Base + 0x100 + 16*n (n = 0,1,2,3,4,5,6,7) Access: Privileged read/write

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	base_address_low_n																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register region_setup_high_n
Offset Instance_Base + 0x104 + 16*n (n = 0,1,2,3,4,5,6,7) Access: Privileged read/write

R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register region_attributes_n
Offset Instance_Base + 0x108 + 16*n (n = 0,1,2,3,4,5,6,7) Access: Privileged read/write

R	sp<n>				0	0	0	0	0	0	0	0	0	0	0	0	subregion_disable<n>								0	size<n>						en<n>
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

64.2.0.1 AHB-TZASC Programming Model Differences vs. (AXI) TZASC

The (AXI) TZASC module supports the notion of read and write speculative accesses. This support reflects, to some degree, to the basic nature and timing requirements of the AXI protocol. At reset, both read and write speculations are enabled and the transaction address and attributes are forwarded to the slave destination *before* the validity of the access has been evaluated.

The TZASC allows AXI data transfers only if the access check allows the reference; if the access check detects a violation, the data transfer is inhibited and the access is redefined as a Denied AXI transaction.

The address speculation is programmable and can be disabled by asserting the appropriate bits in the speculation_control register. There are system performance implications: if address speculation is disabled, then the *TZASC module adds one clock cycle of latency to perform the access check before the address is forwarded* to the slave destination.

By contrast, the *AHB-TZASC does not support, nor need any form of address speculation*. Instead, the module is able to perform the required access check on-the-fly without adding any cycle of latency. If the access check indicates a violation, the AHB-TZASC module is still able to inhibit the reference to the slave.

Accordingly, there are 3 register bits in the TZASC programming model that are specifically related to the address speculation and *not implemented* in the AHB-TZASC module. As highlighted in Figure 64-1, these include:

1) Register bit = lockdown_select[2]

This register bits provides a lock function for speculation_control register, making its content read-only. Since the AHB-TZASC module does not need nor support speculation, this bit is not required and therefore not implemented. This bit is RAZ/WI.

2) Register bits = speculation_control[1:0]

These register bits control the read and write address speculation. Again, since the AHB-TZASC module does not need nor support speculation, these bits are not required and not implemented. These bits are RAO/WI (read as one, write ignored). This signals that speculation is disabled, but as discussed previously, there are *no added cycles of latency.*

In addition to these programming model differences, the AHB-TZASC does *not* implement the integration test registers (itop, itip, itcrg) nor the component configuration registers (periph_id_[4:0], component_id[3:0]).

Chapter 65 On-Chip Memory Controllers

65.1 Overview

The device includes an on-chip ROM (OCROM) controller supporting an 8 MB address space plus three on-chip RAM (OCRAM) memory controllers also supporting an 8 MB address space allocated for this storage type.

The OCRAM address space is logically partitioned into two equal 4 MB spaces: one defined for system RAM (sys) and the other for graphics RAM (gfx). The Faraday architecture spreads 1.5 MB of RAM across three controllers to provide an appropriate amount of data bandwidth to this critical system memory. Additionally, the capabilities associated with these memories vary, based on the required functionality. See Table 65-1 for details.

Table 65-1. On-Chip Map Region Details

Start Address	End Address	Size [Kbytes]	OCMEM Description	Features
On-chip ROM				
0x0000_0000	0x007F_FFFF	8192	OCROM(3)	Boot Memory
On-chip RAM				
0x3F00_0000	0x3F03_FFFF	256	OCRAM0_sys	ECC protected
0x3F04_0000	0x3F07_FFFF	256	OCRAM1_sys	ECC protected
0x3F08_0000	0x3F3F_FFFF	3584	Reserved for Future Use	
0x3F40_0000	0x3F47_FFFF	512	OCRAM2_gfx	Pixel Converter
0x3F48_0000	0x3F4F_FFFF	512	*Configurable* OCRAM2 gfxRAM or, Data Array for L2 Cache	Pixel Converter
0x3F50_0000	0x3F7F_FFFF	3072	Reserved for Future Use	

Each OCRAM*n* controller is a 64-bit AXI implementation, optimized for maximizing data bandwidth with a *3-stage pipelined single-cycle memory structure*. The pipeline stages include: (1) address/write data setup, (2) RAM array access, and (3) read data return. The first pipeline stage overlaps with the last stage of the NIC301 data pipeline.

The remainder of this section provides details on the error correcting code (ECC) associated with the OCRAM_sys memory, the pixel converter associated with the OCRAM_gfx memory and the programming model.

65.1.1 OCRAM_sys Error Correcting Code (ECC)

To support market requirements related to improved functional and transient fault detection capabilities, this device includes ECC support on the system memory portion of the on-chip RAM. This implementation follows the traditional memory ECC design as the check logic is localized to the OCRAM_sys memory controllers.

The implemented ECC is a single error correction, double error detection (SECDED) code using a Hsiao minimum odd weight column definition. The ECC is calculated based on a 64-bit data width. Additionally, the error codes are generated based on more than just the data associated with the storage location in order to protect additional information associated with an access. Specifically, the address corresponding to the access location is combined with the data in the OCRAM_sys controllers to generate error protection codes which cover certain types of addressing errors which may occur in the memory controller or memory array(s). The *8-bit ECC implementation in the OCRAM_sys memory controllers cover the 64 data bits plus the upper 29 bits of the access address*. Details on the H matrix which defines the relationship between the 29-bit address and the 64-bit data fields to produce the 8 checkbits are provided in Section 65.1.1.1, “ECC Checkbit/Syndrome Coding Details”. Note that only single bit errors in the data or checkbit fields can be corrected (on-the-fly); single bit errors pointing to the address field and all detected multi-bit errors are treated as uncorrectable ECC events. Note the terminology uncorrectable or non-correctable are used interchangeably.

The OCRAM_sys ECC function can be summarized in the following steps. Consider reads and writes separately.

For data writes received by the OCRAM_sys memory controller:

- If enabled, the ECC logic generates the 8-bit checkbit field based on the 29-bit address + 64-bit data
- The 64 data bits + 8 checkbits are stored in the memory

For data reads processed by the OCRAM_sys memory controller:

- The memory reads the addressed location returning 64 data bits + 8 checkbits
- If enabled, the ECC logic forms an 8-bit syndrome which is the XOR of the 8 checkbits read from memory and the (re)calculated checkbits formed by passing the 64 data bits (plus the address field checkbits) through the H-matrix logic. An all-zero syndrome indicates an error-free memory read, else a memory corruption event has occurred.
- The ECC logic decodes the syndrome to determine if the event is a 1-bit data error that can be corrected on the fly, or some type of uncorrectable (aka non-correctable) ECC event. If the data or checkbits contains a 1-bit error, it is corrected and returned to the requesting bus master; The device optionally includes the ability to generate an ECC alert interrupt signaling a correctable 1-bit data error occurred. For uncorrectable ECC events, the bus transfer is terminated with an error response, and a separate ECC alert interrupt can be generated. Additionally, the Miscellaneous System Control Module (MSCM) module contains hardware to record the specifics of the uncorrectable ECC event. The Section XREF HERE for details.

Recall the ECC is organized based on a 64-bit data field. Accordingly, the OCRAM_sys memory controllers automatically performs the *required read-modify-write* sequence needed to generate the correct 8 checkbits when a memory write of less than 64 bits is performed. Consider an 8-bit byte write; for this operation, the OCRAM_sys controller reads the 64-bit data location containing the byte to be updated, performs the standard data read ECC evaluation (correcting any single-bit errors) and then merges the new data byte into the 64-bit read data, calculates the new checkbits and writes the entire 64 data bits and 8 checkbits back into the memory. If the read portion of this sequence detects an uncorrectable ECC event, it is immediately signaled, the data write error terminated and the memory update aborted.

The ECC read-modify-write adds two cycles of latency on a less-than-64 bit write: one cycle is needed for the read and a second cycle to perform the ECC operations associated with the read syndrome generation and possible on-the fly correction. The new data is merged with the read data in the “normal” write setup cycle. *Given this 2 cycle latency addition for ECC read-modify-writes, for maximum system performance, writes less than 64 bits should be minimized and/or operated in a decoupled (aka “imprecise”) manner to lessen their impact.*

Note standard ECC operation requires that the entire memory be written during system startup to “initialize” correct checkbit values for all locations before read operations can be checked. To assist in this process, the device implements two independent control flags for each OCRAM_sys controller: one that enables checkbit generation on writes and another to enable checking on read operations. The memory initialization writes can be performed directly by a processor or an appropriately configured DMA channel.

65.1.1.1 ECC Checkbit/Syndrome Coding Details

In the device, ECC scheme implements a single-error correction, double-error detection code using the so-called Hsiao minimum odd-weight column criteria. The Hsiao codes are Hamming distance 4 implementations which provide the SECDED capabilities. The minimum odd-weight constraints defined by Hsiao are relatively simple in the resulting specification of the parity check H matrix which defines the association between the data (and address) bits and the checkbits. They are:

1. There are no all zeroes columns.
2. Every column is distinct.
3. Every column contains an odd number of ones, and hence is “odd weight”.

In defining the H-matrix for this family of devices, these requirements from Hsiao were applied. The resulting ECC is organized based on 64 data bits plus 29 address bits (the upper bits of the 32-bit address field minus the 3 bits which select the byte within 64-bit (8-byte) data field.

The basic H-matrix for this (101, 93) code (93 is the total number of “data” bits (64 data + 29 address), 101 is the total number of data and address bits plus 8 checkbits) is shown in Table 65-10, where an ‘*’ indicates the corresponding data or address bit is XOR’ed to form the final checkbit value on the left. For 64-bit data writes, the table sections corresponding to D[63:32], D[31:0], and A[31:3] are logically summed (output of each table section is XOR’ed) together to the final value driven on the checkbit[7:0] outputs.

Table 65-10. ECC H-Matrix Definition

Checkbits [7:0]	Data Bit[1]																															
	Byte 7								Byte 6								Byte 5								Byte 4							
	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
7	*				*					*			*		*	*		*	*	*	*					*	*			*	*	
6			*	*			*	*					*		*		*				*				*			*	*			*
5	*	*	*	*					*				*	*		*		*	*				*	*	*				*			
4	*		*			*		*					*		*		*			*			*	*			*				*	
3									*	*	*	*	*		*				*		*				*	*	*	*				
2					*		*					*		*			*	*				*	*							*		*
1		*		*		*		*	*	*	*	*						*		*		*		*						*		*
0		*			*	*	*				*			*	*	*		*				*				*		*	*		*	
	Byte 3								Byte 2								Byte 1								Byte 0							
Checkbits [7:0]	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
7					*		*				*		*				*	*			*	*			*				*	*		*
6	*	*		*	*				*				*				*	*		*	*					*	*				*	*
5			*	*			*	*	*			*	*	*					*		*	*		*								
4	*		*	*				*		*	*				*	*	*			*			*	*				*		*		
3	*	*	*			*						*	*		*	*			*				*		*	*	*			*		
2		*		*	*	*	*	*	*	*	*			*					*				*		*		*	*		*		
1		*						*		*			*	*	*			*				*				*			*	*	*	
0		*		*		*		*				*				*				*				*				*	*		*	*
Checkbits [7:0]	Address Bit[1]																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3			
7		*		*		*		*	*	*		*	*	*		*	*	*		*		*		*	*	*		*				
6		*		*	*	*	*			*	*			*	*	*	*		*		*	*		*	*	*		*				
5		*	*	*			*	*			*	*		*		*		*	*			*	*	*		*	*	*	*			
4	*	*	*		*	*		*		*	*	*	*		*					*	*		*	*	*		*	*				
3	*		*		*	*	*		*		*	*	*			*		*		*	*		*		*	*	*		*			
2	*	*			*		*	*	*	*				*	*	*	*	*	*	*		*	*		*		*		*			
1	*		*	*		*	*		*	*		*	*		*		*	*	*		*	*	*					*	*			
0	*		*	*	*			*	*		*		*	*	*		*		*	*	*			*		*	*		*			

1 Bit numbering follows the standard ARM-AMBA little endian convention with bit 0 as LSB: D[7:0] corresponds to byte at address 0, and D[63:56] corresponds to byte at address 7.

Figure 65-1 shows an alternative representation of the ECC encode process, written as a C language function.

Figure 65-1. C Language encodeECC Function Description

```
encodeEcc (addr, data64)
    unsigned int        addr;                          /* 32-bit byte address */
    unsigned long long  data64;                        /* 64-bit data */

{
    unsigned int        addr_ecc;                      /* 8 bits of ecc for address */
    unsigned int        ecc;                           /* 8 bits of ecc codeword */

    /* the following equations calculates the 8-bit wide ecc codeword by examining
    each addr or data bit and xor'ing the appropriate H-matrix value if the bit = 1 */

    addr_ecc
        = (((addr   >> 31) & 1) ? 0x1f : 0x0)      /* addr[31] */
        ^ (((addr   >> 30) & 1) ? 0xf4 : 0x0)      /* addr[30] */
        ^ (((addr   >> 29) & 1) ? 0x3b : 0x0)      /* addr[29] */
        ^ (((addr   >> 28) & 1) ? 0xe3 : 0x0)      /* addr[28] */
        ^ (((addr   >> 27) & 1) ? 0x5d : 0x0)      /* addr[27] */
        ^ (((addr   >> 26) & 1) ? 0xda : 0x0)      /* addr[26] */
        ^ (((addr   >> 25) & 1) ? 0x6e : 0x0)      /* addr[25] */
        ^ (((addr   >> 24) & 1) ? 0xb5 : 0x0)      /* addr[24] */

        ^ (((addr   >> 23) & 1) ? 0x8f : 0x0)      /* addr[23] */
        ^ (((addr   >> 22) & 1) ? 0xd6 : 0x0)      /* addr[22] */
        ^ (((addr   >> 21) & 1) ? 0x79 : 0x0)      /* addr[21] */
        ^ (((addr   >> 20) & 1) ? 0xba : 0x0)      /* addr[20] */
        ^ (((addr   >> 19) & 1) ? 0x9b : 0x0)      /* addr[19] */
        ^ (((addr   >> 18) & 1) ? 0xe5 : 0x0)      /* addr[18] */
        ^ (((addr   >> 17) & 1) ? 0x57 : 0x0)      /* addr[17] */
        ^ (((addr   >> 16) & 1) ? 0xec : 0x0)      /* addr[16] */

        ^ (((addr   >> 15) & 1) ? 0xc7 : 0x0)      /* addr[15] */
        ^ (((addr   >> 14) & 1) ? 0xae : 0x0)      /* addr[14] */
        ^ (((addr   >> 13) & 1) ? 0x67 : 0x0)      /* addr[13] */
        ^ (((addr   >> 12) & 1) ? 0x9d : 0x0)      /* addr[12] */
        ^ (((addr   >> 11) & 1) ? 0x5b : 0x0)      /* addr[11] */
        ^ (((addr   >> 10) & 1) ? 0xe6 : 0x0)      /* addr[10] */
        ^ (((addr   >>  9) & 1) ? 0x3e : 0x0)      /* addr[ 9] */
        ^ (((addr   >>  8) & 1) ? 0xf1 : 0x0)      /* addr[ 8] */

        ^ (((addr   >>  7) & 1) ? 0xdc : 0x0)      /* addr[ 7] */
        ^ (((addr   >>  6) & 1) ? 0xe9 : 0x0)      /* addr[ 6] */
        ^ (((addr   >>  5) & 1) ? 0x3d : 0x0)      /* addr[ 5] */
        ^ (((addr   >>  4) & 1) ? 0xf2 : 0x0)      /* addr[ 4] */
        ^ (((addr   >>  3) & 1) ? 0x2f : 0x0);     /* addr[ 3] */

    ecc = (((data64 >> 63) & 1) ? 0xb0 : 0x0)      /* data[63] */
        ^ (((data64 >> 62) & 1) ? 0x23 : 0x0)      /* data[62] */
        ^ (((data64 >> 61) & 1) ? 0x70 : 0x0)      /* data[61] */
        ^ (((data64 >> 60) & 1) ? 0x62 : 0x0)      /* data[60] */
        ^ (((data64 >> 59) & 1) ? 0x85 : 0x0)      /* data[59] */
        ^ (((data64 >> 58) & 1) ? 0x13 : 0x0)      /* data[58] */
        ^ (((data64 >> 57) & 1) ? 0x45 : 0x0)      /* data[57] */
```

```
^ (((data64 >> 56) & 1) ? 0x52 : 0x0)      /* data[56] */

^ (((data64 >> 55) & 1) ? 0x2a : 0x0)      /* data[55] */
^ (((data64 >> 54) & 1) ? 0x8a : 0x0)      /* data[54] */
^ (((data64 >> 53) & 1) ? 0x0b : 0x0)      /* data[53] */
^ (((data64 >> 52) & 1) ? 0x0e : 0x0)      /* data[52] */
^ (((data64 >> 51) & 1) ? 0xf8 : 0x0)      /* data[51] */
^ (((data64 >> 50) & 1) ? 0x25 : 0x0)      /* data[50] */
^ (((data64 >> 49) & 1) ? 0xd9 : 0x0)      /* data[49] */
^ (((data64 >> 48) & 1) ? 0xa1 : 0x0)      /* data[48] */

^ (((data64 >> 47) & 1) ? 0x54 : 0x0)      /* data[47] */
^ (((data64 >> 46) & 1) ? 0xa7 : 0x0)      /* data[46] */
^ (((data64 >> 45) & 1) ? 0xa8 : 0x0)      /* data[45] */
^ (((data64 >> 44) & 1) ? 0x92 : 0x0)      /* data[44] */
^ (((data64 >> 43) & 1) ? 0xc8 : 0x0)      /* data[43] */
^ (((data64 >> 42) & 1) ? 0x07 : 0x0)      /* data[42] */
^ (((data64 >> 41) & 1) ? 0x34 : 0x0)      /* data[41] */
^ (((data64 >> 40) & 1) ? 0x32 : 0x0)      /* data[40] */

^ (((data64 >> 39) & 1) ? 0x68 : 0x0)      /* data[39] */
^ (((data64 >> 38) & 1) ? 0x89 : 0x0)      /* data[38] */
^ (((data64 >> 37) & 1) ? 0x98 : 0x0)      /* data[37] */
^ (((data64 >> 36) & 1) ? 0x49 : 0x0)      /* data[36] */
^ (((data64 >> 35) & 1) ? 0x61 : 0x0)      /* data[35] */
^ (((data64 >> 34) & 1) ? 0x86 : 0x0)      /* data[34] */
^ (((data64 >> 33) & 1) ? 0x91 : 0x0)      /* data[33] */
^ (((data64 >> 32) & 1) ? 0x46 : 0x0)      /* data[32] */

^ (((data64 >> 31) & 1) ? 0x58 : 0x0)      /* data[31] */
^ (((data64 >> 30) & 1) ? 0x4f : 0x0)      /* data[30] */
^ (((data64 >> 29) & 1) ? 0x38 : 0x0)      /* data[29] */
^ (((data64 >> 28) & 1) ? 0x75 : 0x0)      /* data[28] */
^ (((data64 >> 27) & 1) ? 0xc4 : 0x0)      /* data[27] */
^ (((data64 >> 26) & 1) ? 0x0d : 0x0)      /* data[26] */
^ (((data64 >> 25) & 1) ? 0xa4 : 0x0)      /* data[25] */
^ (((data64 >> 24) & 1) ? 0x37 : 0x0)      /* data[24] */

^ (((data64 >> 23) & 1) ? 0x64 : 0x0)      /* data[23] */
^ (((data64 >> 22) & 1) ? 0x16 : 0x0)      /* data[22] */
^ (((data64 >> 21) & 1) ? 0x94 : 0x0)      /* data[21] */
^ (((data64 >> 20) & 1) ? 0x29 : 0x0)      /* data[20] */
^ (((data64 >> 19) & 1) ? 0xea : 0x0)      /* data[19] */
^ (((data64 >> 18) & 1) ? 0x26 : 0x0)      /* data[18] */
^ (((data64 >> 17) & 1) ? 0x1a : 0x0)      /* data[17] */
^ (((data64 >> 16) & 1) ? 0x19 : 0x0)      /* data[16] */

^ (((data64 >> 15) & 1) ? 0xd0 : 0x0)      /* data[15] */
^ (((data64 >> 14) & 1) ? 0xc2 : 0x0)      /* data[14] */
^ (((data64 >> 13) & 1) ? 0x2c : 0x0)      /* data[13] */
^ (((data64 >> 12) & 1) ? 0x51 : 0x0)      /* data[12] */
^ (((data64 >> 11) & 1) ? 0xe0 : 0x0)      /* data[11] */
^ (((data64 >> 10) & 1) ? 0xa2 : 0x0)      /* data[10] */
^ (((data64 >>  9) & 1) ? 0x1c : 0x0)      /* data[ 9] */
^ (((data64 >>  8) & 1) ? 0x31 : 0x0)      /* data[ 8] */

^ (((data64 >>  7) & 1) ? 0x8c : 0x0)      /* data[ 7] */
```

```
        ^ (((data64 >>  6) & 1) ? 0x4a : 0x0)       /* data[ 6] */
        ^ (((data64 >>  5) & 1) ? 0x4c : 0x0)       /* data[ 5] */
        ^ (((data64 >>  4) & 1) ? 0x15 : 0x0)       /* data[ 4] */
        ^ (((data64 >>  3) & 1) ? 0x83 : 0x0)       /* data[ 3] */
        ^ (((data64 >>  2) & 1) ? 0x9e : 0x0)       /* data[ 2] */
        ^ (((data64 >>  1) & 1) ? 0x43 : 0x0)       /* data[ 1] */
        ^  ((data64         & 1) ? 0xc1 : 0x0);     /* data[ 0] */

    ecc = ecc ^ addr_ecc;   /* combine data and addr ecc values */
    return(ecc);
}
```

As the ECC syndrome is calculated on a read operation by applying the H-matrix to the data plus the checkbits, an all zero syndrome indicates an error free operation. *If the generated syndrome value is non-zero and matches one of the H-matrix values associated with the data or checkbits, it represents a single-bit error correction case and the specific bit is complemented to produce the correct data value*. If the syndrome value matches one of the H-matrix values associated with the address bits, or is an even weight value, or represents an unused odd weight value, a non-correctable ECC event has been detected and the appropriate error termination response is initiated.

Details on the MSCM programming model registers associated with ECC functionality are provided in Section XREF-HERE. This includes capabilities to insert errors to “check the checkers”, both the hardware as well as associated software routines, as well as logic to record bus cycle information at the time of an ECC event.

65.1.1.2 ECC and System Performance Implications

The logic used for the generation of checkbits on writes, calculation of the syndrome and on-the-fly single bit correction on reads is fundamentally integrated into the OCRAM memory controller’s microarchitecture. As a result, the enabling of ECC *does **not** add any machine cycles to RAM reads and 64-bit writes*.

As previously discussed, writes for sizes less than 64 bits require an additional 2 cycles of processing to perform the required read-modify-write to calculate the checkbits needed by the 64-bit ECC algorithm. Additionally, the enabling of the ECC check function requires that the appropriate memory has been completely written to place initialized data in all locations.

65.1.2 OCRAM_gfx Pixel Conversion

The performance of graphic algorithms on a SIMD extension like ARM NEON depends pretty much on the capability to efficiently load and store pixels in the appropriate format. The color components have to be placed in the components of the SIMD register to allow efficient processing.

For byte-aligned pixel format like ARGB8888 this can be easily achieved with the VLD and VST instructions of ARM NEON. While this pixel format offers the full color resolution it requires a significant amount of memory (32 bits per pixel). For single chip solutions without external memory this makes it difficult or even impossible to support larger displays.

One potential solution for this problem is to limit the color depth and spend only 16 bits per pixel for some or all frame buffers / layers. While this halves the amount of memory required for the frame buffers or

layers it is not well suited to be handled with ARM NEON as it requires significant pre-/post-processing to get the pixel components adjusted into the vector components of the SIMD registers.

While implementing the pixel packing and unpacking is a pretty performance intensive task on ARM NEON this is something that can be very easily implemented in a small hardware block located in the memory controller of the on-chip SRAM pool. The purpose of the Pixel Converter is to do the required conversions in a rather simple hardware block."

Accordingly, the Faraday memory controllers include a programmable ability to perform specific pixel conversions based on alternate address spaces mapped to the physical OCRAM_gfx memory. Specifically, the pixel converter hardware reformats memory-resident 16 bit graphics pixel data into 32 bit processor words for the Cortex-A5's NEON SIMD execution.

Graphics algorithms often require simultaneous support for three remapped regions as it is common to have images in two different formats blended together to create a third (output) image. Additionally, the core views the pixel data as "expanded" 32 bit words which is *double the size* of the actual packed 16 bit pixel data in the OCRAM_gfx.

Given the physical OCRAM_gfx space is defined as 4 MB, then the alternate pixel converter view needs 3 regions * (2 * 4 MB/region) = 24 MB as a minimum size. Rounding up, the Faraday memory map allocates 32 MB to this alternate pixel view into the OCRAM_gfx. This alternate pixel view is located in the system address space defined as 0x7E00_0000 - 0x7FFF_FFFF.

65.1.2.1 OCRAM_gfx Pixel Converter Functional Description

The three alternate OCRAM_gfx address spaces are dedicated to specific 16-bit pixel formats. The device architecture supports 3 formats, generically represented as {A}RGB{w}xyz where w defines the number of optional alpha bits, x is the number of red bits, y is the number of green bits and z is the number of blue bits:

- RGB565
- ARGB1555
- ARGB4444

Next, consider the actual data manipulations performed to convert these three 16-bit pixel formats into the 32-bit ARBG8888 format needed by the CA5's NEON SIMD engine. The required conversions are presented both graphically and in C code.

65.1.2.1.1 RGB565

In general, the pixel expansion process involves copying the {x,y,z} bits of the color component into the 8-bit ARGB8888 format left justified and then padding the remaining low-order bits with the appropriate number of most significant bits. The figure below shows the RGB565 conversion into ARGB8888. Throughout this section, let rx {x = 0, 1, 2, 3, 4} define the bits of the red color component, gy {y = 0, 1,

2, 3, 4, 5} be the bits of the green color component and bz {z = 0, 1, 2, 3, 4} the bits of the blue color component.

Figure 65-2. RGB565 → ARGB8888 Conversion

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
r0	r1	r2	r3	r4	g0	g1	g2	g3	g4	g5	b0	b1	b2	b3	b4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	r0	r1	r2	r3	r4	r0	r1	r2	g0	g1	g2	g3	g4	g5	g0	g1	b0	b1	b2	b3	b4	b0	b1	b2

For this conversion, all 8 bits of the alpha byte are set in the AGRB8888 format (0xFF).

65.1.2.1.2 ARGB1555

Using the same notation, the ARGB1555 conversion into ARGB8888 is shown in below.

Figure 65-3. ARGB1555 → ARGB8888 Conversion

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a0	r0	r1	r2	r3	r4	g0	g1	g2	g3	g4	b0	b1	b2	b3	b4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a0	a0	a0	a0	a0	a0	a0	a0	r0	r1	r2	r3	r4	r0	r1	r2	g0	g1	g2	g3	g4	g0	g1	g2	b0	b1	b2	b3	b4	b0	b1	b2

For this conversion, 1 bit alpha value is slewed across the upper byte of the AGRB8888 format.

65.1.2.1.3 ARGB4444

Using the same notation, the ARGB4444 conversion into ARGB8888 is shown in below.

Figure 65-4. ARGB4444 → ARGB8888 Conversion

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a0	a1	a2	a3	r0	r1	r2	r3	g0	g1	g2	g3	b0	b1	b2	b3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a0	a1	a2	a3	a0	a1	a2	a3	r0	r1	r2	r3	r0	r1	r2	r3	g0	g1	g2	g3	g0	g1	g2	g3	b0	b1	b2	b3	b0	b1	b2	b3

65.1.2.1.4 ARGB8888 Conversions to 16-bit Pixel Data

The preceding sections provided details on the bit remapping when converting from the supported 16-bit pixel formats into the 32-bit ARGB8888 format. These conversions are performed on alternate address space *reads* of the OCRAM_gfx memory.

When performing a *write* of 32-bit ARGB8888 pixel data, the conversion process supports two modes of operation, uniquely selectable for each alternate pixel address space. The store modes are *truncation* or *round/clip*.

Truncation simply places the most significant bits for each color component into the available positions in the 16-bit target format.

Figure 65-5. ARGB8888 → ARGB4444 Conversion

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a0	a1	a2	a3	a4	a5	a6	a7	r0	r1	r2	r3	r4	r5	r6	r7	g0	g1	g2	g3	g4	g5	g6	g7	b0	b1	b2	b3	b4	b5	b6	b7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a0	a1	a2	a3	r0	r1	r2	r3	g0	g1	g2	g3	b0	b1	b2	b3

For the truncation operation for this destination format, the low-order 4 bits of each color component, namely a[4-7], r[4-7], g[4-7] and b[4-7] are simply discarded when creating the 16-bit ARGB4444 format.

If the round/clip mode is enabled, the pixel converter first performs two's complement rounding (always round up) by adding a "1" to *the bit immediately to the right of the rounding point*. The rounding bit position is determined by the number of available bits in the given color component of the 16-bit data format; after the addition, the low-order bits to the right of the rounding point are discarded. Specifically, the rounding "1" factor is defined in a C language expression as `1 << (8 - bitwidth - 1)` where `bitwidth` is the size of the color component in the 16-bit pixel data.

After the rounding addition is performed, the result is checked for a carry out. If a carry out is signaled, the data has "overflowed" and the final color component is forced to all ones (the clipped value).

A complete definition of the round/clip functionality, is expressed in the following C code function.

Figure 65-6. Pixel Conversion C Code: Round/Clip Function

```
unsigned char roundclip (unsigned char intensity, int bitwidth)
{
    unsigned int val = intensity;
    val += 1U << (8-bitwidth-1);    /* rounding step */
    if (val & 0x100)                /* check for a cout */
        val = 0xff;                 /* set to clipped value*/

    return (val & 0xff);
}
```

where `intensity` is the source data byte extracted from the ARGB8888 format, and `bitwidth` is the number of bits available in the destination field of the 16-bit pixel data.

From the resulting value only the bitwidth MSBs are used, that is, a truncation step is applied after round/clip arithmetic is calculated.

65.1.2.2 Pixel Conversion and System Performance Implications

Like the ECC implementation, the pixel converter logic is fundamentally integrated into the OCRAM memory controller microarchitecture. As a result, the accesses via the pixel converters *do **not** add any machine cycles to 32- and 64-bit RAM reads and writes*. The conversion on writes is integrated into the first stage (address/write data setup) of the OCRAM's data pipeline and read conversions are handled in the third stage (read data return).

Chapter 66 Miscellaneous System Control Module (MSCM)

66.1 Introduction

The Miscellaneous System Control Module (MSCM) contains Access Control and TrustZone Security hardware, CPU Configuration registers and Interrupt Router control.

66.2 MSCM Access Control and TrustZone Security Memory Map and Register Descriptions

The memory-mapped registers that supports the access control and TrustZone security functions are distributed across multiple modules. In particular, the CSU and all instances of the TZASC controllers are involved along with a portion of the MSCM module. The pertinent MSCM programming model can only be accessed with privileged mode 32-bit references; any other access type or size are terminated with an error.

The MSCM memory map occupies a single 4 KB slave peripheral address space, based at system address 0x4000_1000. The access control and TrustZone security (ACTZS) portion of the MSCM memory map is allocated the upper quarter of the address space and is based at address 0x4000_1C00.

66.2.1 MSCM Access Control and TrustZone Security (ACTZS) Memory Map

The ACTZS configuration portion of the MSCM programming model map is shown in Table 66-1. It is partitioned into two sections:

- Offset addresses 0xC00 - 0xC18 define basic system control and configuration for all the TZASC modules and the CSU.
- Offset addresses 0xD00 - 0xDDC contain captured access address and attribute information for CSL-detected violations. For the logic evaluating the CSLn security levels for all the slave peripheral modules, the *errors are logically combined into one set of reporting registers for each PBRIDGE controller.*

Note, each TZASC module includes local registers within its programming model to record comparable information on TrustZone security violations.

Attempted writes to read-only registers are simply ignored (RO/WI). Privileged mode accesses of the reserved locations at offset addresses 0xC08 - 0xC0C and 0xC1C - 0xCFC are treated as RAZ/WI (read as zero, write ignored). Likewise, privileged mode accesses of the reserved locations at offset addresses 0xD04 + 16*n (n = 0-13) and 0xDE0 - 0xFFC are treated as RAZ/WI (read as zero, write ignored). Attempted user mode accesses in the ACTZS section (offset addresses 0xC00 - 0xDFC) are terminated with an error as are any references with a size smaller than 32 bits.

Table 66-1. MSCM ACTZS Configuration Memory Map

Offset Address	Register Name	Register Description	Width (bits)	Access	Reset Value	Section/Page
0xC00	MSCM_TZENR	ACTZS TrustZone Enable Register	32	R/W	0x0000_0000	66.2.1.1.1/66-3481
0xC04	MSCM_TZIR	ACTZS TrustZone Interrupt Register	32	RO/WI	0x0000_0000	66.2.1.2/66-3483
0xC08-0xC0C		Reserved	32	RAZ/WI	0x0000_0000	–
0xC10	MSCM_CSLIER	ACTZS CSLn Interrupt Enable Register	32	R/W	0x0000_0000	66.2.1.3/66-3484
0xC14	MSCM_CSLIR	ACTZS CSLn Interrupt Register	32	R/W	0x0000_0000	66.2.1.4/66-3485
0xC18	MSCM_CSOVR	ACTZS CSLn Interrupt Overrun Register	32	R/W	0x0000_0000	66.2.1.5/66-3486
0xC1C-0xCFC		Reserved	32	RAZ/WI	0x0000_0000	–
In this device, the following register array is valid for n = [0-13]						
0xD00 + 16*n	MSCM_CSFARn	ACTZS CSLn Fail Status Address (Low) Register	32	RO/WI	0x0000_0000	66.2.1.7/66-3488
0xD04 + 16*n		Reserved	32	RAZ/WI	0x0000_0000	–
0xD08 + 16*n	MSCM_CSFCRn	ACTZS CSLn Fail Status Control Register	32	RO/WI	0x0000_0000	66.2.1.8/66-3489
0xD0C + 16*n	MSCM_CSFIRn	ACTZS CSLn Fail Status Master ID Register	32	RO/WI	0x0000_0000	66.2.1.9/66-3490
0xDE0-0xFFC		Reserved	32	RAZ/WI	0x0000_0000	–

66.2.1.1 MSCM Access Control and TrustZone Security Register Descriptions

This section of the MSCM programming model includes a number of registers controlling the individual TrustZone checkers plus CSLn attribute check logic plus an array of 128-bit register structures containing captured CSLn fault information.

For all the ACTZS registers, privileged 32-bit reads from the CA5, the CM4 or the debugger return the appropriate register information. Reads from any other bus master return all zeroes (RAZ). Privileged writes from the CA5, CM4 or the debugger to writeable registers update the appropriate fields, while privileged writes from other bus masters are ignored (WI). Attempted user mode accesses or any access with a size different than 32 bits are terminated with an error.

66.2.1.1.1 ACTZS TrustZone Enable Register (MSCM_TZENR)

The TrustZone Enable Register is a 32-bit register containing a bit map of enables for each TrustZone Address Space Controller in the device.

Given the AXI-TZASC functionality is complex and can add a cycle of latency to accesses being checked, this register provides the mechanism to completely bypass the module if it is disabled. The register also contains a global TZASC control signal indicating a secure boot process completed and certain registers within the TZASC modules are locked and cannot be altered.

In addition to the TZENR register bits, there is a single device fuse bit that provides a further level of qualification. This fuse signal is combined with the individual TZENi bits to form the fully qualified TZASC enable for each instance:

```
qualified_TZENi =  MSCM_TZENR[i] &  tzenb_fuse
```

where `tzenb_fuse` serves as a global enable qualifier.

For the AHB-TZASC modules, the access checking is performed on-the-fly during the address phase bus cycle. If a given AHB-TZASC module is logically disabled, the internal logic is inhibited as if the `action[reaction_value]` register field is cleared meaning neither bus error terminations nor security violation interrupts are generated.

All the programmable bits in the TZENR are "sticky", that is, once set, they remain set until the next system reset clears all the bits.

The TZENR also contains a lock bit (RO) that may be set to disable writes to the register, preserving the enabled/disabled state of the TZASC modules.

Figure 66-1. ACTZS TrustZone Enable Register (MSCM_TZENR)

Offset 0xC00 — Access: Privileged read/write

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RO	0	0	SBL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TZEN11	0	TZEN9	0	TZEN7	TZEN6	TZEN5	TZEN4	TZEN3	0	TZEN1	0
W	RO			SBL																	TZEN11		TZEN9		TZEN7	TZEN6	TZEN5	TZEN4	TZEN3		TZEN1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(shaded) = Unimplemented or Reserved

Table 66-2. MSCM_TZENR Field Descriptions

Field	Description

31
RO
	Read-Only. This register bit provides a mechanism to “lock” the configuration state defined by MSCM_TZENR. Once asserted, this bit remains set and attempted writes to the MSCM_TZENR register are ignored until the next system reset clears the flag.
	
	`0 = writes to the MSCM_TZENR are allowed`
	`1 = writes to the MSCM_TZENR are ignored`

28
SBL
	Secure Boot Lock. The state of this field is broadcast from MSCM to all the TZASC modules in the device. Once asserted, this bit remains set and it is used by the TZASC modules to lock certain register fields, preventing their contents from modification. Once locked, these special registers cannot be modified until a system reset clears this flag. See the TZASC documentation for details on the specific registers and fields locked by this bit.
	
	`if SBL = 0, then the TZASC’s special register contents are not locked.`
	`if SBL = 1, then the TZASC’s special register contents are locked.`

i
TZENi
	TrustZone Enable (i = 1, 3-7, 9, 11). This field enables/disables each Trust Zone address space controller. Once asserted, the individual bit remains set.
	
	`if TZENi = 0, the TZASC module is disabled and logically bypassed.`
	`if TZENi = 1, the TZASC module is enabled.`
	
	The individual TZENi are mapped to the following TZASC modules:
	
	`1  = FlexBus`
	`3  = CM4-TCM backdoor port`
	`4  = QuadSPI0`
	`5  = OCRAM0_sys`
	`6  = OCRAM1_sys`
	`7  = OCRAM2_gfx`
	`9  = QuadSPI1`
	`11  = DDR0 | DDR1`

66.2.1.2 ACTZS TrustZone Interrupt Register (MSCM_TZIR)

The TrustZone Interrupt Register is a 32-bit read-only register containing a bit map of the interrupt requests from each TrustZone Address Space Controller in the device. This register is intended to provide a simple mechanism for an interrupt service routine to query the state of the TZASC interrupts with a single register read rather than polling all the individual modules.

Once the source of the TrustZone violation interrupt has been determined, the ISR can then query the responsible TZASC for additional details on the security violation.

Attempted privileged mode writes are ignored.

Figure 66-2. ACTZS TrustZone Interrupt Register (MSCM_TZIR)

Offset 0xC04 — Access: Privileged read-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TZINT11	0	TZINT9	0	TZINT7	TZINT6	TZINT5	TZINT4	TZINT3	0	TZINT1	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-3. MSCM_TZIR Field Descriptions

Field	Description

i
TZINTi
	TrustZone Interrupt (i = 1, 3-7, 9, 11). This read-only field signals the assertion of a Trust Zone address space controller interrupt. If a given TZASC's interrupt is asserted, the module can be accessed for more information on the access that generated the security violation.
	
	`if TZINTi = 0, the TZASC module's interrupt is negated.`
	`if TZINTi = 1, the TZASC module's interrupt is asserted.`
	
	The individual TZINTi are mapped to the following TZASC modules:
	
	`1 = FlexBus`
	`3 = CM4-TCM backdoor port`
	`4 = QuadSPI0`
	`5 = OCRAM0_sys`
	`6 = OCRAM1_sys`
	`7 = OCRAM2_gfx`
	`9 = QuadSPI1`
	`11 = DDR0 | DDR1`

66.2.1.3 ACTZS CSLn Interrupt Enable Register (MSCM_CSLIER)

The CSLn Interrupt Enable Register is a 32-bit register containing a bit map of interrupt enables for CSLn logic reporting access check violations.

As the individual CSLn levels are evaluated for each bus transfer, access violations are error terminated and the resulting error status flags collected and posted in the MSCM_CSLIR. This register provides an optional enable to generate a CSLn alert interrupt which is subsequently routed into the CSU's alarm logic.

The CSLIER also contains a lock bit (RO) that may be set to disable writes to the register, preserving the enabled/disabled state of the CSLn interrupts.

Figure 66-3. ACTZS CSLn Interrupt Enable Register (MSCM_CSLIER)

Offset 0xC10 | Access: Privileged read/write

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CIE13	CIE12	0	CIE10	CIE9	CIE8	0	0	0	CIE4	CIE3	CIE2	CIE1	CIE0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-4. MSCM_CSLIER Field Descriptions

Field	Description

31
RO
	Read-Only. This register bit provides a mechanism to “lock” the configuration state defined by MSCM_CSLIER. Once asserted, attempted writes to the MSCM_CSLIER register are ignored until the next system reset clears the flag.
	
	`0 = writes to the MSCM_CSLIER are allowed`
	`1 = writes to the MSCM_CSLIER are ignored`

i
CIEi
	CSLn Interrupt Enable (i = 0-4, 8-10, 12-13). This field enables/disables each CSLn access violation alert interrupt.
	
	`if CIEi = 0, the CSLn access check interrupt is disabled.`
	`if CIEi = 1, the CSLn access check interrupt is  enabled.`
	
	The individual CIEi are mapped to the following CSLn modules (or accumulated access check errors):
	
	` 0  = Boot ROM`
	` 1  = FlexBus`
	` 2  = GPIO`
	` 3  = CM4-TCM backdoor port`
	` 4  = QuadSPI0`
	` 8  = SecureRAM`
	` 9  = QuadSPI1`
	`10  = RLE`
	`12  = PBRIDGE0 (logical summation of CSL[12- 72])`
	`13  = PBRIDGE1 (logical summation of CSL[73-120])`

66.2.1.4 ACTZS CSLn Interrupt Register (MSCM_CSLIR)

The CSLn Interrupt Register is a 32-bit register containing a bit map of interrupts for CSLn logic reporting access check violations.

As the individual CSLn levels are evaluated for each bus transfer, access violations are error terminated and the resulting error status flags collected and posted in the MSCM_CSLIR. This register is then combined with the MSCM_CSLIER to optionally generate a CSLn alert interrupt which is subsequently routed into the CSU’s alarm logic.

The MSCM_CSLIR records all CSLn access violations regardless of the state of the interrupt enable (MSCM_CSLIER). Accordingly, this register can be used by both interrupt service routines and/or bus error exception handlers to quickly determine the source of the CSLn access check violation. Each individual interrupt flag in this register is cleared by writing a “1” to it; this would typically be done after the captured fail address and attribute information is retrieved from the appropriate MSCM_CSF*R register. Additionally, the clearing of an interrupt flag in this register also clears the corresponding bit in the MSCM_CSLOVR register and rearms the logic for capturing the failed access information.

Figure 66-4. ACTZS CSLn Interrupt Register (MSCM_CSLIR)

Offset 0xC14 — Access: Privileged read/write

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	INT13	INT12	0	INT10	INT9	INT8	0	0	0	INT4	INT3	INT2	INT1	INT0
W																			w1c	w1c		w1c	w1c	w1c				w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(Shaded) = Unimplemented or Reserved

Table 66-5. MSCM_CSLIR Field Descriptions

Field	Description

i
INTi
	CSLn Interrupt (i = 0-4, 8-10, 12-13). This field records each CSLn access violation alert interrupt.
	
	`if INTi = 0, the CSLn access check interrupt is negated.`
	`if INTi = 1, the CSLn access check interrupt is asserted.`
	
	The individual INTi are mapped to the following CSLn modules (or accumulated access check errors):
	
	`0 = Boot ROM`
	`1 = FlexBus`
	`2 = GPIO`
	`3 = CM4-TCM backdoor port`
	`4 = QuadSPI0`
	`8 = SecureRAM`
	`9 = QuadSPI1`
	`10 = RLE`
	`12 = PBRIDGE0 (logical summation of CSL[12- 72])`
	`13 = PBRIDGE1 (logical summation of CSL[73-120])`

66.2.1.5 ACTZS CSLn Interrupt Overrun Register (MSCM_CSOVR)

The CSLn Interrupt Overrun Register is a 32-bit read-only register containing a bit map of “overrun” interrupt conditions for the CSLn logic reporting access check violations.

The overrun condition is simply defined as the detection of another CSLn access check violation before the previous one has been full processed and cleared. Stated differently, if a CSLn access check violation is detected and the corresponding MSCM_CSLIR[i] bit still asserted, the MSCM_CSOVR[i] bit is set.

Additionally, for the slave ACTZS ports using the AXI bus protocol, it is possible to detect *simultaneous* access violations on the read and write channels. For this condition, the CSLn Fail Status Registers (MSCM_CSFARn, MSCM_CSFCRn, MSCM_CSFIRn) capture the information associated with the write access and set the appropriate overrun indicator in the MSCM_CSOVR.

Each individual interrupt flags is cleared by writing a “1” to it and this is typically be done after the captured fail address and attribute information is retrieved from the appropriate MSCM_CSF*R register. The clearing of an interrupt flag in the MSCM_CSLIR also *clears the corresponding bit in the MSCM_CSOVR register and rearms the logic for capturing the fail access information*.

Figure 66-5. ACTZS CSLn Interrupt Overrun Register (MSCM_CSOVR)

Offset 0xC18 Access: Privileged read-only

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVR13	OVR12	0	OVR10	OVR9	OVR8	0	0	0	OVR4	OVR3	OVR2	OVR1	OVR0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-6. MSCM_CSOVR Field Descriptions

Field	Description

i
OVRi
	CSLn Interrupt Overrun (i = 0-4, 8-10, 12-13). This field records each CSLn access violation interrupt overrun condition, where another security violation is detected before the previous one has been fully processed and cleared.
	
	`if OVRi = 0, a CSLn access check overrun has not been detected.`
	`if OVRi = 1, a CSLn access check overrun has been detected.`
	
	The individual OVRi are mapped to the following CSLn modules (or accumulated access check errors):
	
	` 0  = Boot ROM`
	` 1  = FlexBus`
	` 2  = GPIO`
	` 3  = CM4-TCM backdoor port`
	` 4  = QuadSPI0`
	` 8  = SecureRAM`
	` 9  = QuadSPI1`
	`10  = RLE`
	`12  = AIPS0 (logical summation of CSL[12- 72])`
	`13  = AIPS1 (logical summation of CSL[73-120])`

66.2.1.6 ACTZS CSLn Fail Status Capture Registers

This section of the MSCM_ACTZS programming model contains an array of four word (128-bit) data values containing address and attribute information corresponding to CSLn access check violations. The

format of this data structure is identical to the fail status information captured by the TZASC when they detect a security violation.

When a CSLn access check violation is detected, the bus transaction is error terminated, the appropriate bit in the MSCM_CSLIR set and the fail address and attribute information captured in the corresponding data structure. The contents of the captured fail data is unaffected until the interrupt flag is cleared by writing a 1 to it, at which time, the capturing of fail information is rearmed.

The device supports n = [0-13] and contains an array of ten 128-bit data structures and four reserved structures as defined in Table 66-7.

Table 66-7. MSCM CSLn Fail Status Capture Registers

Base Offset Address	Source
0xD00	Boot ROM
0xD10	FlexBus
0xD20	GPIO
0xD30	CM4-TCM backdoor port
0xD40	QuadSPI0
0xD50-0xD70	– (Reserved)
0xD80	SecureRAM
0xD90	QuadSPI1
0xDA0	RLE
0xDB0	– (Reserved)
0xDC0	PBRIDGE0 = Logical Summation of CSR[12- 72]
0xDD0	PBRIDGE1 = Logical Summation of CSR[73-120]
0xDE0 - 0xFFC	– (Reserved)

66.2.1.7 ACTZS CSLn Fail Status Address (Low) Register (MSCM_CSFARn)

The CSFARn is a 32-bit read-only register for capturing the low-order 32 bits of the address of the last captured access check violation detected by the CSLn logic. When the CSLn logic detects an access violation, the address and attributes associated with the memory reference are loaded into the CSFARn, CSFCRn and CSFIRn registers, and the appropriate flag in the CSLn Interrupt Register (CSLIR) is asserted.

For the slave ACTZS ports using the AXI bus protocol, it is possible to detect *simultaneous* access violations on the read and write channels. For this condition, the CSLn Fail Status Registers (MSCM_CSFARn, MSCM_CSFCRn, MSCM_CSFIRn) capture the information associated with the write access and additionally set the appropriate overrun indicator in the MSCM_CSOVR.

The contents of the MSCM_CSFARn is unaffected until the corresponding MSCM_CSLIR interrupt flag is cleared by writing a 1 to it, at which time, the capturing of fail information is rearmed.

Attempted privileged mode writes are ignored.

NOTE: The *next* sequential word at offset address 0xD04 + 16*n is reserved for systems where the address space is larger than 4 Gbytes. In the device, this read-only word is always hardwired to zero.

Figure 66-6. ACCTZS Fail Status Address (Low) Register (MSCM_CSFARn)

Offset 0xD00 + 16**n* (MSCM_CSFAR, *n* = 0-13) Access: Privileged read-only

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FAD																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-8. MSCM_CSFARn Field Descriptions

Field	Description
31–0 FAD	CSLn Fail Address. This read-only field specifies the system address from the last captured CSLn access check violation.

66.2.1.8 ACTZS CSLn Fail Status Control Register (MSCM_CSFCRn)

The CSFCRn is a 32-bit read-only register for capturing specific attribute bits of the last captured access check violation detected by the CSLn logic. When the CSLn logic detects an access violation, the address and attributes associated with the memory reference are loaded into the CSFARn, CSFCRn and CSFIRn registers, and the appropriate flag in the CSLn Interrupt Register (CSLIR) is asserted.

For the slave ACTZS ports using the AXI bus protocol, it is possible to detect *simultaneous* access violations on the read and write channels. For this condition, the CSLn Fail Status Registers (MSCM_CSFARn, MSCM_CSFCRn, MSCM_CSFIRn) capture the information associated with the write access and additionally set the appropriate overrun indicator in the MSCM_CSOVR.

The contents of the MSCM_CSFCRn is unaffected until the corresponding MSCM_CSLIR interrupt flag is cleared by writing a 1 to it, at which time, the capturing of fail information is rearmed.

Attempted privileged mode writes are ignored.

Figure 66-7. ACCTZS Fail Status Control Register (MSCM_CSFCRn)

Offset 0xD08 + 16**n* (MSCM_CSFCR, *n* = 0-13) Access: Privileged read-only

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	FWT	0	0	FNS	FPR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-9. MSCM_CSFCRn Field Descriptions

Field	Description

24
FWT
	CSLn Fail Write. This read-only field specifies the read/write attribute from the last captured CSLn access check violation.
	
	`if FWT = 0, then the last captured CSLn access check violation was a read.`
	`if FWT = 1, then the last captured CSLn access check violation was a write.`

21
FNS
	CSLn Fail Nonsecure. This read-only field specifies the secure/nonsecure attribute from the last captured CSLn access check violation.
	
	`if FNS = 0, the last captured CSLn access check violation was a    secure access.`
	`if FNS = 1, the last captured CSLn access check violation was a nonsecure access.`

20
FPR
	CSLn Fail Privileged. This read-only field specifies the user/privileged attribute from the last captured CSLn access check violation.
	
	`if FPR = 0, the last captured CSLn access check violation was a user mode access.`
	`if FPR = 1, the last captured CSLn access check violation was a privileged access.`

66.2.1.9 ACTZS CSLn Fail Status Master ID Register (MSCM_CSFIRn)

The CSFIRn is a 32-bit read-only register for capturing the master ID attribute of the last captured access check violation detected by the CSLn logic. When the CSLn logic detects an access violation, the address and attributes associated with the memory reference are loaded into the CSFARn, CSFCRn and CSFIRn registers, and the appropriate flag in the CSLn Interrupt Register (CSLIR) is asserted.

For the slave ACTZS ports using the AXI bus protocol, it is possible to detect *simultaneous* access violations on the read and write channels. For this condition, the CSLn Fail Status Registers (MSCM_CSFARn, MSCM_CSFCRn, MSCM_CSFIRn) capture the information associated with the write access and additionally set the appropriate overrun indicator in the MSCM_CSOVR.

The contents of the MSCM_CSFIRn is unaffected until the corresponding MSCM_CSLIR interrupt flag is cleared by writing a 1 to it, at which time, the capturing of fail information is rearmed.

Attempted privileged mode writes are ignored.

Figure 66-8. ACCTZS Fail Status Master ID Register (MSCM_CSFIRn)

Offset 0xD0C + *n* (MSCM_CSFIR, *n* = 0-13) Access: Privileged read-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FMID				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-10. MSCM_CSFIRn Field Descriptions

Field	Description
4–0 FMID	CSLn Fail Master ID. This read-only field specifies the master ID from the last captured CSLn access check violation. The master port numbers are defined in Table 4-2.

66.3 Chip Configuration and Boot

The device configuration is defined by e-fuse bits, supported memory sizes and packing options. Collectively, these configuration bits define an RCON (reset configuration) value, input to the platform as a signal named *num_cores[1:0]*. The supported processor configurations are detailed in Table 66-11.

Table 66-11. Processor Core Configurations

num_cores[1:0]	Core Configuration
0	Cortex-A5 core only
1	Cortex-M4 core only
2	Dual core with Cortex-A5 as CP0 (boot core), Cortex-M4 as CP1
3	Dual core with Cortex-M4 as CP0 (boot core), Cortex-A5 as CP1

By default, both cores are intended to boot from a ROM based at system address 0 although the specifics of exception processing, *including reset, differ between the processors*. For the device, the exception vector table for the Cortex-A5 core is based at system address 0 while the first two words of the exception vector table for the Cortex-M4 are remapped (by the platform hardware) to system addresses 0x0001_6000 and 0x0001_6004; these two vectors correspond to the initial stack pointer value and the initial program counter respectively. The platform hardware remaps the *first two non-debug code fetches after reset is negated* from the Cortex-M4 core as follows:

- CodeBus address 0x0000_0000 is translated to 0x0001_6000 (initial stack pointer)
- CodeBus address 0x0000_0004 is translated to 0x0001_6004 (initial program counter)

By convention, the remapping of the CM4 reset vectors to these addresses implies the exception vector table for this core is based at system address 0x0001_6000.

Once the cores have fetched the needed reset vector(s), it is expected they read core and system configuration information from a globally-accessible slave peripheral that properly converts the *num_cores[1:0]* information into more appropriate values. More specifically, the cores access configuration information from a *common set of peripheral addresses* and the chip configuration logic properly evaluates based on the requesting processor and returns the appropriate value for the given processor including core identification including presence or absence of optional hardware capabilities, local memory sizes, logical processor number for multi-core configurations, bus master number, etc.

As an example, there is a single 32-bit read-only location for the core identification: a 32-bit read from this location returns a four character ASCII string: 0x43_41_35_01 (“CA5”) or 0x43_4D_34_01” (“CM4”) depending on the requesting master.

Given this approach, the processors can efficiently execute from a single Boot ROM image and pass control to the appropriate sections in the system startup code.

The programming model associated with the core configuration information is included as part of the Miscellaneous System Control Module (MSCM). It specifically includes multiple views of the processor configuration; one that is available generically to the cores and others that are available to any bus masters in the system.

66.3.1 MSCM CPU Configuration Memory Map and Register Descriptions

The CPU configuration portion of the MSCM module provides a set of memory-mapped read-only addresses defining the processor setup. This portion of the MSCM programming model can only be accessed with privileged mode 32-bit read references; any other access type or size are terminated with an error. If the processor is logically *not included* in the chip configuration, reads of its configuration registers return zeroes.

The MSCM memory map occupies a single 4 Kbyte slave peripheral address space, based at system address 0x4000_1000.

66.3.1.1 MSCM CPU Configuration Memory Map

The CPU configuration portion of the MSCM programming model map is shown in Table 66-12. This read-only portion of the MSCM memory map is organized based on the *logical processor number* (not any type of physical port number) and partitioned into three equal sections:

- Offset addresses 0x000 - 0x01F define the generic processor “x” configuration information. This region is only accessible to the CA5 and CM4 processor cores; reads by non-core bus masters (including the debugger) are treated as RAZ (read as zero) accesses.
- Offset addresses 0x020 - 0x03F define the configuration information for processor 0 (CP0). This region is accessible to any bus master.
- Offset addresses 0x040 - 0x05F define the configuration information for processor 1 (CP1). This region is accessible to any bus master. For uniprocessor chip configurations, reads of this section are treated as RAZ.

Privileged mode accesses of the reserved locations at offset addresses 0x060 - 0x3FC are treated as RAZ (read as zero). Attempted user mode or write accesses in the CPU configuration section (offset addresses 0x000 - 0x3FC) are terminated with an error.

Table 66-12. MSCM CPU Configuration Memory Map

Offset Address	Register Name	Register Description	Width (bits)	Access	Reset Value	Section/Page
0x000	MSCM_CPxTYPE	Processor x Type Register	32	RO	f(*num_cores*)	66.3.1.2.1/66-3495
0x004	MSCM_CPxNUM	Processor x Number Register	32	RO	f(*num_cores*)	66.3.1.2.2/66-3495
0x008	MSCM_CPxMASTER	Processor x Master Number Register	32	RO	f(*num_cores*)	66.3.1.2.3/66-3496
0x00C	MSCM_CPxCOUNT	Processor x Count Register	32	RO	f(*num_cores*)	66.3.1.2.4/66-3497
0x010	MSCM_CPxCFG0	Processor x Configuration Register 0	32	RO	f(*num_cores*)	66.3.1.2.5/66-3497
0x014	MSCM_CPxCFG1	Processor x Configuration Register 1	32	RO	f(*num_cores*)	66.3.1.2.6/66-3498

Table 66-12. MSCM CPU Configuration Memory Map

Offset Address	Register Name	Register Description	Width (bits)	Access	Reset Value	Section/Page
0x018	MSCM_CPxCFG2	Processor x Configuration Register 2	32	RO	f(*num_cores*)	66.3.1.2.7/66-3499
0x01C	MSCM_CPxCFG3	Processor x Configuration Register 3	32	RO	f(*num_cores*)	66.3.1.2.8/66-3500
0x020	MSCM_CP0TYPE	Processor 0 Type Register	32	RO	f(*num_cores*)	66.3.1.2.1/66-3495
0x024	MSCM_CP0NUM	Processor 0 Number Register	32	RO	f(*num_cores*)	66.3.1.2.2/66-3495
0x028	MSCM_CP0MASTER	Processor 0 Master Number Register	32	RO	f(*num_cores*)	66.3.1.2.3/66-3496
0x02C	MSCM_CP0COUNT	Processor 0 Count Register	32	RO	f(*num_cores*)	66.3.1.2.4/66-3497
0x030	MSCM_CP0CFG0	Processor 0 Configuration Register 0	32	RO	f(*num_cores*)	66.3.1.2.5/66-3497
0x034	MSCM_CP0CFG1	Processor 0 Configuration Register 1	32	RO	f(*num_cores*)	66.3.1.2.6/66-3498
0x038	MSCM_CP0CFG2	Processor 0 Configuration Register 2	32	RO	f(*num_cores*)	66.3.1.2.7/66-3499
0x03C	MSCM_CP0CFG3	Processor 0 Configuration Register 3	32	RO	f(*num_cores*)	66.3.1.2.8/66-3500
0x040	MSCM_CP1TYPE	Processor 1 Type Register	32	RO	f(*num_cores*)	66.3.1.2.1/66-3495
0x044	MSCM_CP1NUM	Processor 1 Number Register	32	RO	f(*num_cores*)	66.3.1.2.2/66-3495
0x048	MSCM_CP1MASTER	Processor 1 Master Number Register	32	RO	f(*num_cores*)	66.3.1.2.3/66-3496
0x04C	MSCM_CP1COUNT	Processor 1 Count Register	32	RO	f(*num_cores*)	66.3.1.2.4/66-3497
0x050	MSCM_CP1CFG0	Processor 1 Configuration Register 0	32	RO	f(*num_cores*)	66.3.1.2.5/66-3497
0x054	MSCM_CP1CFG1	Processor 1 Configuration Register 1	32	RO	f(*num_cores*)	66.3.1.2.6/66-3498
0x058	MSCM_CP1CFG2	Processor 1 Configuration Register 2	32	RO	f(*num_cores*)	66.3.1.2.7/66-3499
0x05C	MSCM_CP1CFG3	Processor 1 Configuration Register 3	32	RO	f(*num_cores*)	66.3.1.2.8/66-3500
0x060-0x3FC		Reserved	32	RAZ	0x0000_0000	–

66.3.1.2 MSCM CPU Configuration Register Descriptions

66.3.1.2.1 Processor x Type Register

The MSCM_CPxTYPE register provides a CPU-specific response indicating the personality of the core making the access. The 32 bit response includes 3 ASCII characters defining the CPU type (“CA5” or “CM4”) along with a byte defining the logical revision number. The logical revision number follows ARM’s rYpZ nomenclature.

A privileged read from the CA5 or the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Figure 66-9. Processor x Type Register (MSCM_CPxType)

Offset 0x000 (MSCM_CPxTYPE) — Access: Privileged read-only

Bits	31–8	7–0
R	Personality	rYpZ
W	(unimplemented)	(unimplemented)
Reset	- (each bit)	- (each bit)

= Unimplemented or Reserved

Table 66-13. MSCM_CPxTYPE Field Descriptions

Field	Description

31–8
Personality
	Processor x Personality. This 24-bit read-only field defines the processor personality for CPx:
	
	`if CPx = Cortex-A5, then Personality = 0x43_41_35 (“CA5”).`
	`if CPx = Cortex-M4, then Personality = 0x43_4D_34 (“CM4”).`

7–0
rYpZ
	Processor x Revision. This 8-bit read-only field defines the processor revision for CPx:
	
	`if CPx = Cortex-A5, then rYpZ = 0x01 corresponding to the r0p1 core release.`
	`if CPx = Cortex-M4, then rYpZ = 0x01 corresponding to the r0p1 core release.`

66.3.1.2.2 Processor x Number Register (MSCM_CPxNUM)

The MSCM_CPxNUM register provides a CPU-specific response indicating the logical processor number of the core making the access. In single processor configurations, the logical processor number is always zero; in dual core configurations, the boot (or primary) core is assigned number 0 while the secondary core is defined as number 1.

A privileged read from the CA5 or the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Figure 66-10. Processor x Number Register (MSCM_CPxNUM)

Offset 0x004 (MSCM_CPxNUM) Access: Privileged read-only

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CPN
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-

= Unimplemented or Reserved

Table 66-14. MSCM_CPxNUM Field Descriptions

Field	Description

0
CPN
	Processor x Number. This zero-filled word defines the logical processor number for CPx:
	
	`if single core configuration, then CPN = 0`
	`   else if dual core configuration && boot (primary) core, then CPN = 0`
	`        else secondary core, CPN = 1`

66.3.1.2.3 Processor x Master Register

The MSCM_CPxMASTER register provides a CPU-specific response indicating the physical bus master number the core making the access. The 32 bit response defines the physical master number for processor x.

A privileged read from the CA5 or the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Figure 66-11. Processor x Master Register (MSCM_CPxMaster)

Offset 0x008 (MSCM_CPxMASTER) Access: Privileged read-only

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PPN				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	-	-

= Unimplemented or Reserved

Table 66-15. MSCM_CPxMASTER Field Descriptions

Field	Description

4–0
PPN
	Processor x Physical Port Number. This 5-bit read-only field defines the physical port number for CPx. The value is device specific.
	
	`For the Cortex-A5 core in this device, PPN = 0x2`
	`For the Cortex-M4 core in this device, PPN = 0x0`

66.3.1.2.4 Processor x Count Register

The MSCM_CPxCOUNT register provides a CPU-specific response indicating the total number of processor cores in the chip configuration. For this device, the count value is 0 or 1, depending on whether the configuration is a single (0) or dual (1) processor.

A privileged read from the CA5 or the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Figure 66-12. Processor x Count Register (MSCM_CPxCOUNT)

Offset 0x00C (MSCM_CPxCOUNT) — Access: Privileged read-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PCNT
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-

= Unimplemented or Reserved

Table 66-16. MSCM_CPxCOUNT Field Descriptions

Field	Description

0
PCNT
	Processor Count. This 1-bit read-only field defines the processor count for the chip configuration:
	`if single core configuration, then PCNT = 0`
	`   else if dual core configuration, then PCNT = 1`

66.3.1.2.5 Processor x Configuration 0 Register

The MSCM_CPxCFG0 register provides a CPU-specific response detailing configuration information, in this case, information on the Level 1 caches (if present).

A privileged read from the CA5 or the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Figure 66-13. Processor x Configuration 0 Register (MSCM_CPxCFG0)

Offset 0x010 (MSCM_CPxCFG0) — Access: Privileged read-only

Bit	31–24	23–16	15–8	7–0
R	ICSZ	ICWY	DCSZ	DCWY
W				
Reset	- - - - - - - -	- - - - - - - -	- - - - - - - -	- - - - - - - -

= Unimplemented or Reserved

Table 66-17. MSCM_CPxCFG0 Field Descriptions

Field	Description

31–24
ICSZ
	Level 1 Instruction Cache Size. This 8-bit read-only field provides an encoded value of the Instruction Cache size. The capacity of the memory is expressed as Size [bytes] = $2^{(8+SZ)}$ where SZ is non-zero; a SZ = 0 indicates the memory is not present.
	
	`if no          Instruction Cache, then ICSZ = 0x00`
	`if a   4 Kbyte Instruction Cache, then ICSZ = 0x04`
	`if an  8 Kbyte Instruction Cache, then ICSZ = 0x05`
	`if a  16 Kbyte Instruction Cache, then ICSZ = 0x06`
	`if a  32 Kbyte Instruction Cache, then ICSZ = 0x07`
	`if a  64 Kbyte Instruction Cache, then ICSZ = 0x08`
	`if a 128 Kbyte Instruction Cache, then ICSZ = 0x09`
	`if a 256 Kbyte Instruction Cache, then ICSZ = 0x0A`
	`if a 512 Kbyte Instruction Cache, then ICSZ = 0x0B`
	
	`For the Cortex-A5 core in this device, ICSZ = 0x07 (32 Kbytes)`
	`For the Cortex-M4 core in this device, ICSZ = 0x06 (16 Kbytes)`

23–16
ICWY
	Level 1 Instruction Cache Ways. This 8-bit read-only field provides the number of cache ways for the Instruction Cache.
	
	`For the Cortex-A5 core in this device, ICWY = 0x02 (2-way set-associative)`
	`For the Cortex-M4 core in this device, ICWY = 0x02 (2-way set-associative)`

15–8
DCSZ
	Level 1 Data Cache Size. This 8-bit read-only field provides an encoded value of the Data Cache size. The capacity of the memory is expressed as Size [bytes] = $2^{(8+SZ)}$ where SZ is non-zero; a SZ = 0 indicates the memory is not present.
	
	`if no          Data Cache, then DCSZ = 0x00`
	`if a   4 Kbyte Data Cache, then DCSZ = 0x04`
	`if an  8 Kbyte Data Cache, then DCSZ = 0x05`
	`if a  16 Kbyte Data Cache, then DCSZ = 0x06`
	`if a  32 Kbyte Data Cache, then DCSZ = 0x07`
	`if a  64 Kbyte Data Cache, then DCSZ = 0x08`
	`if a 128 Kbyte Data Cache, then DCSZ = 0x09`
	`if a 256 Kbyte Data Cache, then DCSZ = 0x0A`
	`if a 512 Kbyte Data Cache, then DCSZ = 0x0B`
	
	`For the Cortex-A5 core in this device, DCSZ = 0x07 (32 Kbytes)`
	`For the Cortex-M4 core in this device, DCSZ = 0x06 (16 Kbytes)`

23–16
DCWY
	Level 1 Data Cache Ways. This 8-bit read-only field provides the number of cache ways for the Data Cache.
	
	`For the Cortex-A5 core in this device, DCWY = 0x04 (4-way set-associative)`
	`For the Cortex-M4 core in this device, DCWY = 0x02 (2-way set-associative)`

66.3.1.2.6 Processor x Configuration 1 Register

The MSCM_CPxCFG1 register provides a CPU-specific response detailing configuration information, in this case, information on a Level 2 cache (if present).

A privileged read from the CA5 or the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Figure 66-14. Processor x Configuration 1 Register (MSCM_CPxCFG1)

Offset 0x014 (MSCM_CPxCFG1) Access: Privileged read-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	L2SZ								L2WY								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																																
Reset	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-18. MSCM_CPxCFG1 Field Descriptions

Field	Description

31–24
L2SZ
	Level 2 Cache Size. This 8-bit read-only field provides an encoded value of the Level 2 Cache size. The capacity of the memory is expressed as Size [bytes] = $2^{(8+SZ)}$ where SZ is non-zero; a SZ = 0 indicates the memory is not present.
	
	`if no          Level 2 Cache, then L2SZ = 0x00`
	`if a    4 Kbyte Level 2 Cache, then L2SZ = 0x04`
	`if an   8 Kbyte Level 2 Cache, then L2SZ = 0x05`
	`if a   16 Kbyte Level 2 Cache, then L2SZ = 0x06`
	`if a   32 Kbyte Level 2 Cache, then L2SZ = 0x07`
	`if a   64 Kbyte Level 2 Cache, then L2SZ = 0x08`
	`if a 128 Kbyte Level 2 Cache, then L2SZ = 0x09`
	`if a 256 Kbyte Level 2 Cache, then L2SZ = 0x0A`
	`if a 512 Kbyte Level 2 Cache, then L2SZ = 0x0B`
	
	`For the Cortex-A5 core in this device,`
	`    if L2 is present, then L2SZ = 0x0B (512 Kbytes)`
	`       else L2SZ = 0x00 (not present)`
	`For the Cortex-M4 core in this device, L2SZ = 0x00 (not present)`

23–16
L2WY
	Level 2 Cache Ways. This 8-bit read-only field provides the number of cache ways for the Level 2 Cache.
	
	`For the Cortex-A5 core in this device,`
	`    if L2 is present, then L2WY = 0x08 (8-way set-associative)`
	`       else L2WY = 0x00 (not present)`
	`For the Cortex-M4 core in this device, L2WY = 0x00 (not present)`

66.3.1.2.7 Processor x Configuration 2 Register

The MSCM_CPxCFG2 register provides a CPU-specific response detailing configuration information, in this case, information on tightly-coupled local memories (if present).

A privileged read from the CA5 or the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Figure 66-15. Processor x Configuration 2 Register (MSCM_CPxCFG2)

Offset 0x018 (MSCM_CPxCFG2) Access: Privileged read-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TMLSZ								0	0	0	0	0	0	0	1	TMUSZ								0	0	0	0	0	0	0	1
W																																
Reset	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	1	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	1

= Unimplemented or Reserved

Table 66-19. MSCM_CPxCFG2 Field Descriptions

Field	Description

31–24
TMLSZ
	Tightly-coupled Memory Lower Size. This 8-bit read-only field provides an encoded value of the tightly-coupled local memory lower size. The capacity of the memory is expressed as Size [bytes] = $2^{(8+SZ)}$ where SZ is non-zero; a SZ = 0 indicates the memory is not present.
	
	`if no TCML, then TMLSZ = 0x00`
	`if a 4 Kbyte TCML, then TMLSZ = 0x04`
	`if an 8 Kbyte TCML, then TMLSZ = 0x05`
	`if a 16 Kbyte TCML, then TMLSZ = 0x06`
	`if a 32 Kbyte TCML, then TMLSZ = 0x07`
	`if a 64 Kbyte TCML, then TMLSZ = 0x08`
	`if a 128 Kbyte TCML, then TMLSZ = 0x09`
	`if a 256 Kbyte TCML, then TMLSZ = 0x0A`
	`if a 512 Kbyte TCML, then TMLSZ = 0x0B`
	
	`For the Cortex-A5 core in this device, TMLSZ = 0x00 (not present)`
	`For the Cortex-M4 core in this device, TMLSZ = 0x07 (32 Kbytes)`

15–8
TMUSZ
	Tightly-coupled Memory Upper Size. This 8-bit read-only field provides an encoded value of the tightly-coupled local memory upper size. The capacity of the memory is expressed as Size [bytes] = $2^{(8+SZ)}$ where SZ is non-zero; a SZ = 0 indicates the memory is not present.
	
	`if no TCMU, then TMUSZ = 0x00`
	`if a 4 Kbyte TCMU, then TMUSZ = 0x04`
	`if an 8 Kbyte TCMU, then TMUSZ = 0x05`
	`if a 16 Kbyte TCMU, then TMUSZ = 0x06`
	`if a 32 Kbyte TCMU, then TMUSZ = 0x07`
	`if a 64 Kbyte TCMU, then TMUSZ = 0x08`
	`if a 128 Kbyte TCMU, then TMUSZ = 0x09`
	`if a 256 Kbyte TCMU, then TMUSZ = 0x0A`
	`if a 512 Kbyte TCMU, then TMUSZ = 0x0B`
	
	`For the Cortex-A5 core in this device, TMUSZ = 0x00 (not present)`
	`For the Cortex-M4 core in this device, TMUSZ = 0x07 (32 Kbytes)`

66.3.1.2.8 Processor x Configuration 3 Register

The MSCM_CPxCFG3 register provides a CPU-specific response detailing configuration information, in this case, information on processor options.

A privileged read from the CA5 or the CM4 returns the appropriate processor information. Reads from any other bus master return all zeroes. Attempted user mode or write accesses are terminated with an error.

Figure 66-16. Processor x Configuration 3 Register (MSCM_CPxCFG3)

Offset 0x01C (MSCM_CPxCFG3) Access: Privileged read-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SBP		0	BB	CMP	TZ	MMU	JAZ	SIMD	FPU
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	0	-	-	-	-	-	-	-

= Unimplemented or Reserved

Table 66-20. MSCM_CPxCFG3 Field Descriptions

Field	Description

9–8
SBP
	System Bus Ports. This 2-bit read-only field defines the number of physical connections to the system bus fabric for this processor.
	
	`For the Cortex-A5 core in this device, SBP = 0x1`
	`For the Cortex-M4 core in this device, SBP = 0x2`

6
BB
	Bit Banding. This 1-bit read-only field defines if the processor supports “bit banding”:
	
	`if bit banding is not supported, then BB = 0x0`
	`if bit banding is     supported, then BB = 0x1`
	
	`For the Cortex-A5 core in this device, BB = 0x0 (not supported)`
	`For the Cortex-M4 core in this device, BB = 0x0 (not supported)`

5
CMP
	Core Memory Protection unit. This 1-bit read-only field indicates if the core memory protection hardware is included in the processor.
	
	`if core memory protection is not included, then CMP = 0x0`
	`if core memory protection is     included, then CMP = 0x1`
	
	`For the Cortex-A5 core in this device, CMP = 0x0 (not supported)`
	`For the Cortex-M4 core in this device, CMP = 0x0 (not supported)`

4
TZ
	Trust Zone. This 1-bit read-only field indicates if the Trust Zone capabilities are supported in the processor.
	
	`if Trust Zone support is not included, then TZ = 0x0`
	`if Trust Zone support is     included, then TZ = 0x1`
	
	`For the Cortex-A5 core in this device, TZ = 0x1 (supported)`
	`For the Cortex-M4 core in this device, TZ = 0x0 (not supported)`

Field	Description

3
MMU
	Memory Management Unit. This 1-bit read-only field indicates if the virtual memory management capabilities are supported in the processor.
	
	`if MMU support is not included, then MMU = 0x0`
	`if MMU support is     included, then MMU = 0x1`
	
	`For the Cortex-A5 core in this device, MMU = 0x1 (supported)`
	`For the Cortex-M4 core in this device, MMU = 0x0 (not supported)`

2
JAZ
	Jazelle. This 1-bit read-only field indicates if Jazelle hardware is supported in the processor.
	
	`if Jazelle support is not included, then JAZ = 0x0`
	`if Jazelle support is     included, then JAZ = 0x1`
	
	`For the Cortex-A5 core in this device, JAZ = 0x0 (not supported)`
	`For the Cortex-M4 core in this device, JAZ = 0x0 (not supported)`

1
SIMD
	SIMD/NEON instruction support. This 1-bit read-only field indicates if the instruction set extensions supporting SIMD and/or NEON capabilities are supported in the processor.
	
	`if SIMD/NEON support is not included, then SIMD = 0x0`
	`if SIMD/NEON support is     included, then SIMD = 0x1`
	
	`For the Cortex-A5 core in this device, SIMD = 0x1 (supported)`
	`For the Cortex-M4 core in this device, SIMD = 0x1 (supported)`

0
FPU
	Floating Point Unit. This 1-bit read-only field indicates if hardware support for floating point capabilities are supported in the processor.
	
	`if FPU support is not included, then FPU = 0x0`
	`if FPU support is     included, then FPU = 0x1`
	
	`For the Cortex-A5 core in this device, FPU = 0x1 (supported)`
	`For the Cortex-M4 core in this device, FPU = 0x1 (supported)`

66.3.1.2.9 MSCM_CP0<name> Registers

The 8 word registers with offset addresses 0x020 - 0x03F define the configuration information for processor 0 (CP0). These registers have exactly the same bit field formats of the corresponding MSCM_CPx<name> registers. See Table 66-12 for the appropriate cross references between the MSCM_CP0<name> and MSCM_CPx<name> registers.

A privileged read from *any bus master* returns the appropriate processor information. Attempted user mode or write accesses are terminated with an error.

66.3.1.2.10 MSCM_CP1<name> Registers

The 8 word registers with offset addresses 0x040 - 0x05F define the configuration information for processor 1 (CP1). These registers have exactly the same bit field formats of the corresponding MSCM_CPx<name> registers. See Table 66-12 for the appropriate cross references between the MSCM_CP1<name> and MSCM_CPx<name> registers.

A privileged read from *any bus master* returns the appropriate processor information. Attempted user mode or write accesses are terminated with an error. For uniprocessor chip configurations, reads of this section are treated as RAZ.

66.3.2 MSCM Interrupt Router Memory Map and Register Descriptions

The interrupt router portion of the MSCM module provides a set of memory-mapped registers defining the interrupt routing and directed processor interrupts. This portion of the MSCM programming model can only be accessed with *privileged mode references from the processor cores or the debugger*; any user mode reference is terminated with an error and all privileged accesses from non-core (and non-debug) masters are treated as RAZ/WI (read as zero, write ignored) references. Additionally, the access size must match the register size (including the reserved spaces) else an error is generated. On attempted read accesses that are error terminated, the returned read data is undefined; on attempted write accesses that are error terminated, the operation is aborted and the destination register unaffected.

Recall the MSCM memory map occupies a single 4 Kbyte slave peripheral address space, based at system address 0x4000_1000.

66.3.2.1 MSCM Interrupt Router Memory Map

The interrupt router portion of the MSCM programming model map is shown in Table 66-21. This portion of the MSCM memory map is organized based on the *logical processor number* (not any type of physical number) and partitioned into two sections:

- Offset addresses 0x800 - 0x820 define the directed CPU interrupt requests
- Offset addresses 0x880 - 0x95E define the interrupt router control for the system

Privileged mode accesses of the reserved locations at offset addresses 0x824 - 0x87C and 0x960 - 0xFFE are treated as RAZ/WI (read as zero, write ignored).

Table 66-21. MSCM Interrupt Router Memory Map

Offset Address	Register Name	Register Description	Width (bits)	Access	Reset Value	Section/Page
0x800	MSCM_IRCP0IR	Interrupt Router CP0 Interrupt Register	32	RW	0x0000_0000	66.3.2.2.1/66-3505
0x804	MSCM_IRCP1IR	Interrupt Router CP1 Interrupt Register	32	RW	0x0000_0000	66.3.2.2.2/66-3506
0x808 – 0x81C		Reserved	32	RAZ/WI	0x0000_0000	–
0x820	MSCM_IRCPGIR	Interrupt Router CPU Generate Interrupt Register	32	WO	–	66.3.2.2.3/66-3507
0x824 – 0x87C		Reserved	32	RAZ/WI	0x0000_0000	–
0x880 – 0x95E	MSCM_IRSPRC*n*, n = 0 – 127	Interrupt Router Shared Peripheral Routing Control *n*	16	RW	0x0000	66.3.2.2.4/66-3508
0x960 – 0xFFE		Reserved	16	RAZ/WI	0x0000	–

66.3.2.2 MSCM Interrupt Router Register Descriptions

66.3.2.2.1 Interrupt Router CP0 Interrupt Register

The MSCM_IRCP0IR register provides an interrupt bit map where each bit defines the state of a unique CPU interrupt. CPU interrupts are asserted by the appropriate write to the MSCM_IRCPGIR with the appropriate interrupt routing bit enabled; see Section 66.3.2.2.3, "Interrupt Router CPU Generate Interrupt Register". A processor interrupt is cleared in an interrupt service routine by writing a "1" to the appropriate bit in the MSCM_IRCP0IR register. Only the CA5, CM4 or debugger can clear its associated interrupt. Privileged accesses from non-core bus masters are treated as RAZ/WI and any attempted user mode reference is terminated with an error. Attempted accesses using a size different than a 32-bit word are also error terminated.

Figure 66-17. Interrupt Router CP0 Interrupt Register (MSCM_IRCP0IR)

Offset 0x800 (MSCM_IRCP0IR) Access: Privileged read/write

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	INT3	INT2	INT1	INT0
W																													w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-22. MSCM_IRCP0IR Field Descriptions

Field	Description

3
INT3
	Interrupt 3. This register bit generates directed interrupt 3 to processor 0 if the appropriate interrupt routing bit is enabled:
	
	`0 = no interrupt is asserted`
	`1 = interrupt 3 to CP0 is asserted`
	
	The interrupt is asserted through an appropriate write to the MSCM_IRCPGIR register. The interrupt is negated when a processor or the debugger writes a "1" to this bit.

2
INT2
	Interrupt 2. This register bit generates directed interrupt 2 to processor 0 if the appropriate interrupt routing bit is enabled:
	
	`0 = no interrupt is asserted`
	`1 = interrupt 2 to CP0 is asserted`
	
	The interrupt is asserted through an appropriate write to the MSCM_IRCPGIR register. The interrupt is negated when a processor or the debugger writes a "1" to this bit.

Field	Description

1
INT1
	Interrupt 1. This register bit generates directed interrupt 1 to processor 0 if the appropriate interrupt routing bit is enabled:
	
	`0 = no interrupt is asserted`
	`1 = interrupt 1 to CP0 is asserted`
	
	The interrupt is asserted through an appropriate write to the MSCM_IRCPGIR register. The interrupt is negated when a processor or the debugger writes a “1” to this bit.

0
INT0
	Interrupt 0. This register bit generates directed interrupt 0 to processor 0 if the appropriate interrupt routing bit is enabled:
	
	`0 = no interrupt is asserted`
	`1 = interrupt 0 to CP0 is asserted`
	
	The interrupt is asserted through an appropriate write to the MSCM_IRCPGIR register. The interrupt is negated when a processor or the debugger writes a “1” to this bit.

66.3.2.2.2 Interrupt Router CP1 Interrupt Register

The MSCM_IRCP1IR register provides an interrupt bit map where each bit defines the state of a unique CPU interrupt. CPU interrupts are asserted by the appropriate write to the MSCM_IRCPGIR with the appropriate interrupt routing bit enabled; see Section 66.3.2.2.3, “Interrupt Router CPU Generate Interrupt Register”. A processor interrupt is cleared in an interrupt service routine by writing a “1” to the appropriate bit in the MSCM_IRCP1IR register. Only the CA5, CM4 or debugger can clear its associated interrupt. Privileged accesses from non-core bus masters are treated as RAZ/WI and any attempted user mode reference is terminated with an error. Attempted accesses using a size different than a 32-bit word are also error terminated.

This register’s functionality and format are equivalent to MSCM_IRCP0IR except the directed interrupts are targeted at CP1.

Figure 66-18. Interrupt Router CP1 Interrupt Register (MSCM_IRCP1IR)

Offset 0x804 (MSCM_IRCP1IR) Access: Privileged read/write

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	INT3	INT2	INT1	INT0
W																													w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-23. MSCM_IRCP1IR Field Descriptions

Field	Description

3
INT3
	Interrupt 3. This register bit generates directed interrupt 3 to processor 1 if the appropriate interrupt routing bit is enabled:
	
	`0 = no interrupt is asserted`
	`1 = interrupt 3 to CP1 is asserted`
	
	The interrupt is asserted through an appropriate write to the MSCM_IRCPGIR register. The interrupt is negated when a processor or the debugger writes a "1" to this bit.

2
INT2
	Interrupt 2. This register bit generates directed interrupt 2 to processor 1 if the appropriate interrupt routing bit is enabled:
	
	`0 = no interrupt is asserted`
	`1 = interrupt 2 to CP1 is asserted`
	
	The interrupt is asserted through an appropriate write to the MSCM_IRCPGIR register. The interrupt is negated when a processor or the debugger writes a "1" to this bit.

1
INT1
	Interrupt 1. This register bit generates directed interrupt 1 to processor 1 if the appropriate interrupt routing bit is enabled:
	
	`0 = no interrupt is asserted`
	`1 = interrupt 1 to CP1 is asserted`
	
	The interrupt is asserted through an appropriate write to the MSCM_IRCPGIR register. The interrupt is negated when a processor or the debugger writes a "1" to this bit.

0
INT0
	Interrupt 0. This register bit generates directed interrupt 0 to processor 1 if the appropriate interrupt routing bit is enabled:
	
	`0 = no interrupt is asserted`
	`1 = interrupt 0 to CP1 is asserted`
	
	The interrupt is asserted through an appropriate write to the MSCM_IRCPGIR register. The interrupt is negated when a processor or the debugger writes a "1" to this bit.

66.3.2.2.3 Interrupt Router CPU Generate Interrupt Register

The *write-only* MSCM_IRCPGIR register provides the mechanism for processors to generate directed CPU interrupts. Processor writes to this register provide an *indirect mechanism* to assert the processor interrupts contained in the MSCM_IRCP*n*IR registers. A directed CPU interrupt also requires that the appropriate routing enable is asserted. This register follows the format of the GIC's ICDSGIR (used for software generated interrupts). Privileged writes from non-core (and non-debug) bus masters are treated

as WI and any attempted user mode or read reference is terminated with an error. Attempted accesses using a size different than a 32-bit word are also error terminated.

Figure 66-19. Interrupt Router CPU Generate Interrupt Register (MSCM_IRCPGIR)

Offset 0x820 (MSCM_IRCPGIR) Access: Privileged write-only

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W							TLF								CPUTL																INTID	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-24. MSCM_IRCPGIR Field Descriptions

Field	Description

25–24
TLF
	Target List Field. This 2-bit write-only field is used to assert directed CPU interrupts.
	
	`00  Use the CPUTL (CPU Target List) field to assert directed CPU interrupt(s)`
	`01  Assert directed CPU interrupts for all processors` *`except`* `the requesting core`
	`10  Assert the directed CPU interrupt for` *`only`* `the requesting core`
	`11  Reserved`

17–16
CPUTL
	CPU Target List. This 2-bit write-only field is used when the TLF = 00, and defines the destination directed CPU interrupts to be asserted.
	
	`if CPUTL[17] = 1, then MSCM_IRCP1IR[INTID] is asserted`
	`if CPUTL[16] = 1, then MSCM_IRCP0IR[INTID] is asserted`

1–0
INTID
	Interrupt ID. This 2-bit write-only field defines the destination MSCM_IRCP*n*IR interrupts to be asserted.
	
	`if INTID[1:0] = 00, then MSCM_IRCPnIR[0] loaded as defined by TLF & CPUTL`
	`if INTID[1:0] = 01, then MSCM_IRCPnIR[1] loaded as defined by TLF & CPUTL`
	`if INTID[1:0] = 10, then MSCM_IRCPnIR[2] loaded as defined by TLF & CPUTL`
	`if INTID[1:0] = 11, then MSCM_IRCPnIR[3] loaded as defined by TLF & CPUTL`

66.3.2.2.4 Interrupt Router Shared Peripheral Routing Control Register *n*

The MSCM_IRSPRC*n* register provides an array of halfword (16 bit) registers, where each register defines the routing control for the corresponding interrupt request. In this device, each interrupt request can be routed to neither (0), either (1, 2) or both (3) cores using the processor's logical number. If all the CPxEn bits are cleared, the interrupt request is disabled. Each routing control halfword can be locked by asserting the RO bit. Privileged accesses from non-core (and non-debug) bus masters are treated as

RAZ/WI and any attempted user mode reference is terminated with an error. Attempted accesses using a size different than a 16-bit halfword are also error terminated.

Figure 66-20. Interrupt Router Shared Peripheral Routing Control Register (MSCM_IRSPRC*n*)

Offset 0x880 + 2**n* (MSCM_IRSPRC*n*, *n* = 0-111) Access: Privileged read/write

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RO	0	0	0	0	0	0	0	0	0	0	0	0	0	CP1En	CP0En
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 66-25. MSCM_IRSPRC*n* Field Descriptions

Field	Description

15
RO
	Read-Only. This register bit provides a mechanism to “lock” the routing of the corresponding interrupt request. Once asserted, attempted writes to the MSCM_IRSPRCn register are ignored until the next reset clears the flag.
	
	`0 = writes to the MSCM_IRSPRCn are allowed`
	`1 = writes to the MSCM_IRSPRCn are ignored`

1
CP1En
	Enable CP1 Interrupt. This register bit enables the routing of the corresponding interrupt request to CP1:
	
	`0 = routing to CP1 for the corresponding interrupt request is disabled`
	`1 = routing to CP1 for the corresponding interrupt request is enabled`
	
	This bit cannot be set in uniprocessor configurations.

0
CP0En
	Enable CP0 Interrupt. This register bit enables the routing of the corresponding interrupt request to CP0:
	
	`0 = routing to CP0 for the corresponding interrupt request is disabled`
	`1 = routing to CP0 for the corresponding interrupt request is enabled`

Appendix A
Revision History of this Document

This appendix describes corrections to the Vybrid Reference Manual For convenience. Grammatical and formatting changes are not listed here unless the meaning of something changed.

A.1 Changes between revision 5 and 4

Table A-1. Changes between revisions 5 and 4

Chapter	Description
Gloabl Changes	Editorial Updates throughout the document

A.2 Changes between revisions 4 and 3

Table A-2. Changes between revisions 4 and 3

Chapter	Description
Gloabl Changes between Rev 4 and Rev 3	Removed the content related to Security specific feature of the device. To refer to the Security specific features of the device, refer to the device Security Refernce Manual. • Removed the CAAM chapter, SNVS, Temperature and Voltage Monitor, Security Architecture • Removed security specific content from the device features in Overview chapter • Removed security specific content from the memory map • Removed the registers that are related to security • Removed security specific content from the System Boot chapter Removed instances of Doze mode
Introduction	• In the Vybrid Platform, updated the features
Memory Map	• Added a footnote below memory map • In Peripheral Bridge 0 (AIPS-Lite 0) Memory Map, added footnotes about ARM references

Chapter	Description
Signal Multiplexing	Updated pin muxing table with the following changes: • Added MII0 including MAC0.TXDATA[2], MAC0.TXDATA[3], MAC0.RXDATA[2], MAC0.RXDATA[3] , MAC0.TXERR, MAC0.TXCLK, MAC0.RXCLK, MAC0.COL, MAC0.CRS • Following signals muxed on same RMII0 Pins : MII0_MDC, MII0_MDC, MII0_RXD[1], MII0_RXD[0], MII0_RXER, MII0_TXD[1], MII0_TXD[0], MII0_TXEN • Replaced FB_ALE with FB_MUXED_ALE, FB_CS4_b with FB_MUXED_TSIZ0, FB_TSIZ1 with FB_MUXED_TSIZ1, FB_TBST_b with FB_MUXED_TBST_b, FB_BE0_b with FB_MUXED_BE0_b • Removed RCON18,19,20 • Replaced ESAI_SDO2 with ESAI_SDO2/ESAI_SDI3 • Replaced ESAI_SDO3 with ESAI_SDO3/ESAI_SDI2 • Replaced ESAI_SDI0 with ESAI_SDO5/ESAI_SDI0 • Replaced ESAI_SDI1 with ESAI_SDO4/ESAI_SDI1 • CKO1 additionally muxed at PAD40
IOMUXC	• In the Overview section, added a note • In the register memory map, removed Dummy QUADSPI0 Pad Control Register and Dummy QUADSPI1 Pad Control Register • In the IOMUXC_PTA6, IOMUXC_PTA9, IOMUXC_PTA21, IOMUXC_PTC0, IOMUXC_PTB18, IOMUXC_PTC0-IOMUXC_PTC8, IOMUXC_PTD16-IOMUXC_PTD23 Mux Mode field description updated • Revised the DUMMY PADS (DDR/QuadSPI) section
Chip Configuration	• In the Cortex-M4 Processor Core, added notes • In the Cortex-A5 Processor Core, added a note • In the Interrupt Assignment section., updated rows with 0x0000_01B4 and 0x0000_01B8 • In the PDB implementation with ADC section, updated the figure • In the Ethernet Subsystem, added MII and RMII configuration section • In the Ethernet Subsystem. added note in the Instantiation section
Clocking Overview	• Clarified the Clock Generation and Distribution. Clarified the notes below the diagram. • Clarified Clock sources section • Revised and clarified the PLL Summary section • Updated the PLL properties section • Updated the PLL Features subsection and reaaranged them • Clarified the Typical PLL Configuration table • Minor edits in theTypical PFD Configuration table • In Clock Modes section, updated the Synchronous Mode table • Updated the Asynchronous DDR mode table • In the Ethernet RMII/MII Clocking section, added a note
Clock Controller Module (CCM)	In the feature section: • PLL7 definition added in the table • "Editorial changes In the CCM_CCR regsiter, modified table title in FXOSC_EN bit field description. In the CCM_CLPCR register: • Added "FXOSC_BYPSEN" write-only bit at [10] offset • Added "FXOSC_BYPSS" read-only bit at [12] offset • In CCR[OSCNT] description, changed "FXOSC" to "SXOSC"

Chapter	Description
Analog components control digital interface (ANADIG)	• Minor edits completed in the Regsiter description • In the PLL2 Control[DIV_SELECT], corrected the bit position • Updated the ANADIG_REG_3P0 reset value • In the ANADIG_PLL1_CTRL register, corrected the bit position of DIV_SELECT
Slow Clock Source Controller Module (SCSC)	In the SCSC_SIRC_CTR[SIRC_DIV] field, added a note (TKT106993)
GPC	• Editorial updates
System Reset Controller(SRC)	• Editorial updates • Updated SRC Boot Mode register (SBMR1) field description. • Updated SRC Status register (SRC_SRSR) description by adding Notes and updating the Reset value to 0x00FEFF65
System Boot	• Editorial updates in the chapter • Updated the Boot Flow figure • Updated the BOOT eFuse Descriptions table with RCON 16 entry • Added the two rows at the end of the BOOT eFuse Descriptions table • Editorial updates in the Normal Frequency Clocks Configuration • Clarified the QuadSPI Configuration Parameters table • Added the FlexCAN Configuration Parameters section • In the Flash Control Block Structure table, added • In the NAND Boot eFUSE Descriptions table, added last two rows
Local Memory Controller	• In the Cache control register (LMEM_PCCCR) register, added PCCR2 and PCCR3
Quad Serial Peripheral Interface (QuadSPI)	• Updated "QuadSPI Block Diagram" by updating IOFA[3:0] and IOFB[3:0] to IOFA[7:0] and IOFB[7:0] respectively. • Updated Figure "Serial Flash Access Scheme". • In section "Flash Read/Data Transfer from the QuadSPI Module Internal Buffers" added a figure for "QuadSPI memory map". • In Table " Instruction set" updated Number of Pins for 2' d3 from "Reserved" to "Eight pads" • Updated"QSPI_FR[IUEF]" bit description. • Added a new section"HBURST Support" in "Functional Support". • Added a new Table "Byte Ordering in Parallel Flash Mode- TX Buffer" in Section "Byte Ordering of Serial Flash Read Data". • Added a new Section "Data Strobe Signal functionality".
LPDDR2/DDR3 SDRAM Memory Controller (DDRMC)	• Added a note in Control Register 57(DDRMC_CR57)/Bit CTRL_RAW "Before using the ECC mode the memory needs to be initialized such that all ECC bits are computed and stored in memory." • Added a new section“Initialization of memory when using ECC” under “ECC Options” in Functional Description”. • Updated CR_154 register by adding DDR_SEL_PAD_Contr and DDR_SEL_PAD_Contr bits. • Update CR_155 register by interchanging position for PAD_ODT_BYTE0 and PAD_ODT_BYTE1.
On-Chip One Time Programmable (OCOTP) Controller	• Updated Table: Fuse map address • Updated OCOTP_LOCK register (bit 7) • Updated OCOTP_CFG4 register • Updated OCOTP_VTD_TRIM register • Updated OCOTP_VTMON register

Chapter	Description
Universal Serial Bus (USB) Controller	• Updated section: Normal Mode • Updated section: Low Power Mode • Updated Non Core register section • Updated USBCx_CTRL register (bit 12, 11, and 10) • Updated USBx_PORTSC1 register • Updated section USB Power Control Block • Updated section: USB core interrupts • Updated Table: Device/Host capability registers • Removed OTG operations section • Updated section: Host Controller Initialization • Added a NOTE to section "Write Back qTD" • Updated section: Device Controller Initialization
Universal Serial Bus 2.0 Integrated PHY (USBPHY)	• In the USB PHY Transmitter Control Register, added the bit that controls edge rate • Changed the reset value of USB_ANALOG_USB0_CHRG_DETECT and USB_ANALOG_USB1_CHRG_DETECT
Integrated Interchip Sound (I2S) / Synchronous Audio Interface (SAI)	• In the "Bit clock" section of the "Clocking" section, added a new final paragraph: "If the SAI transmitter or receiver is using an externally generated bit clock in asynchronous mode and that bit clock is generated by an SAI that is disabled in stop mode, then the transmitter or receiver should be disabled by software before entering stop mode. This issue does not apply when the transmitter or receiver is in a synchronous mode because all synchronous SAIs are enabled and disabled simultaneously. (TKT060581) • In the descriptions of TCR2[MSEL] and RCR2[MSEL], added this Note: "Depending on the device, some Master Clock options might not be available. See the chip configuration details for the availability and chip-specific meaning of each option.
Asynchronous Sample Rate Converter (ASRC)	• Edited the field description of ASRC_ASRMCRA[INFIFO_THRESHOLDA] • Edited the field description of ASRC_ASRMCRC
Display Control Unit (DCU4)	• Data layout for RGB565 format,ARGB4444 format,ARGB1555 formatted. • Editorial updates • Updated Conditional tags to remove instance of "PDI interface"
Video subsytem	• Added NOTE to VDEC_MANOVR register • AFE_OFFDRV register, bit field input_pulldown_en [7:4], description updated • AFE_INPFLT register, bit field mux_enlf [6:3], description updated

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: VYBRIDRM
Rev. 5
07/2013